

Calibration and Image Reconstruction Techniques for FMCW MIMO Radars with Consideration of Antenna Radiation Patterns

Kalibrations- und Bildrekonstruktionsverfahren für
FMCW-MIMO-Radare unter Berücksichtigung der
Antennenabstrahlungscharakteristik

Daniel Gotzens

Masterarbeit vorgelegt dem



Institut für
Hochfrequenz-
technik



Lehrstuhl für Radarsystemtechnik
Institut für Hochfrequenztechnik
RWTH Aachen University
February 26, 2024

In Kooperation mit



indurad GmbH Aachen

Abstract

Increased image quality can be achieved by implementing backprojection on a millimeter-wave TDM FMCW MIMO Radar. In this thesis, a backprojection-based imaging algorithm is proposed and implemented for a radar sensor. This radar sensor's functionality is tested, the sensor is calibrated, and then the algorithm is compared to the current FFT-based algorithm as well as MUSIC. It is found that near-field accuracy of the image generated with backprojection and MUSIC is increased, and that real-time capability can only be guaranteed for backprojection and the current FFT-based approach.

Contents

1	Introduction	3
1.1	Radar Applications in Mining	3
1.2	MIMO Radar Sensors	4
1.3	Imaging Algorithms for 3D MIMO Radar Sensors	5
1.4	Thesis Outline	6
2	Radar Fundamentals: Theoretical Background	7
2.1	Antenna Fundamentals	7
2.1.1	Physical Background	7
2.1.2	Antenna Characteristics	7
2.2	Signal Model	7
2.2.1	Single Channel FMCW	8
2.2.2	Uniform Linear Arrays	9
2.2.3	Virtual Arrays	9
2.2.4	Multiplexing Techniques	9
2.3	Image Reconstruction	10
2.3.1	Discrete Fourier Transform	10
2.3.2	Backprojection	12
2.3.3	MUSIC	12
3	Hardware Evaluation	14
3.1	Indurad Multi-Channel Radar	14
3.1.1	Sytem Parameters	15
3.1.2	Virtual Antenna Array	15
3.2	Stability Analysis	16
3.2.1	Setup	16
3.2.2	Preprocessing and Analysis	18
3.2.3	Effects of System Temperature and Runtime	18
3.2.4	Effects of Self-Calibration	18
3.2.5	Antenna Separation	19
3.2.6	Differences between Antennas	19
3.2.7	Conclusion	23
3.3	Antenna Characteristics	24
3.3.1	Setup	24

3.3.2	Preprocessing and Analysis	25
3.3.3	Range Estimation	25
3.3.4	Amplitude	28
3.3.5	Phase Estimation Techniques	32
3.3.6	Phase Analysis	33
3.3.7	Antenna Separation	34
3.3.8	Conclusion	37
4	Image Reconstruction	39
4.1	PyTorch	39
4.2	FFT-Based Imaging	39
4.2.1	Offline Calibration	40
4.2.2	Direction of Arrival Accuracy	40
4.2.3	Result	43
4.3	Backprojection Imaging	43
4.3.1	Direction of Arrival Accuracy	44
4.3.2	Result	46
4.4	Hybrid Approach	46
4.4.1	Direction of Arrival Accuracy	46
4.5	Conclusion	46
5	Conclusions	47
6	Discussion and Outlook	48
A	Code	49
A.1	Introduction to PyTorch Sensor Processing	49
A.2	Listings	51
B	Additional Experiments	54

Chapter 1

Introduction

Radar technology has undergone significant advancements over the decades, revolutionizing various fields including aerospace, defense, automotive, and healthcare. Its fundamental principle of using electromagnetic waves to detect the presence, direction, distance, and speed of objects has enabled myriad applications, ranging from weather monitoring to target tracking.

Within this expansive domain, Frequency Modulated Continuous Wave (FMCW) Multiple Input Multiple Output (MIMO) Millimeter Wave (mmWave) radar systems have emerged as a cutting-edge solution with unparalleled capabilities. FMCW radar, characterized by its continuous transmission of frequency-modulated signals, combined with MIMO architecture, which utilizes multiple antennas for both transmission and reception, has enabled enhanced spatial resolution, improved target detection, and increased resilience to interference. Operating in the mmWave spectrum, typically within the frequency range of 24 to 100 GHz, these radars offer advantages such as high resolution, immunity to environmental conditions like fog and precipitation, and the ability to accommodate large bandwidths for high data rates.

The fusion of FMCW, MIMO, and mmWave technologies represents a significant leap forward in radar sensing, promising to address the evolving demands of modern applications ranging from autonomous vehicles to biomedical imaging. This thesis endeavors to contribute to this dynamic field by proposing and evaluating novel imaging algorithms optimized for FMCW MIMO mmWave radar systems, with a focus on enhancing their performance, efficiency, and applicability across diverse scenarios and applications.

1.1 Radar Applications in Mining

Radar technology finds diverse applications across various industries, with one notable area being mining. In the mining sector, radar systems play a crucial role in enhancing safety, efficiency, and productivity.

Radar-based collision avoidance systems are employed in mining operations to prevent accidents involving heavy machinery and personnel. These systems utilize radar sensors to detect nearby objects and provide real-time alerts to operators, facilitating safe maneuvering in challenging environments.

Moreover, radar technology is instrumental in various logistical aspects of mining operations. It is utilized for stockpile management, enabling accurate measurement and monitoring of ore and waste material reserves, optimizing inventory control and resource allocation. Radar sensors are also integrated into silo and train loading processes, ensuring efficient and precise transfer of bulk materials onto transport vehicles. Furthermore, radar-based systems facilitate ship unloading and berthing operations at port facilities, streamlining the handling of bulk commodities. One significant advantage of radar systems over visual-based solutions in mining environments is their resilience to omnipresent dust and grime. Unlike optical systems, radar sensors are not affected by adverse weather conditions or obstructed visibility, ensuring reliable performance and continuous operation even in harsh mining environments.

1.2 MIMO Radar Sensors

In the realm of radar sensing, various approaches exist for achieving multidimensionality in imaging. Traditional radar systems employ one-dimensional (1D) sensors, which provide information along a single axis, typically range. These sensors are adept at measuring distances to targets but lack detailed spatial information. To overcome this limitation, two-dimensional (2D) sensors have been developed, capable of scanning in both range and azimuth dimensions, providing a more comprehensive view of the surrounding environment. These sensors often achieve multidimensionality either mechanically, by utilizing mechanisms such as rotating antennas or electronically with phased arrays to sweep the radar beam across the scene. Additionally, three-dimensional (3D) radar sensors offer even greater spatial resolution by adding an elevation dimension to the azimuth and range measurements. These sensors are commonly used in applications requiring detailed volumetric imaging. The mechanical movement or rotation of the sensor enables it to scan the scene from multiple angles, capturing information in three dimensions.

The evolution of radar technology has led to the development of advanced imaging techniques that leverage the principles of Multiple Input Multiple Output (MIMO) radar systems. These systems, characterized by their use of multiple transmit and receive antennas, offer significant improvements in imaging performance compared to traditional radar architectures. One key advantage of MIMO radar is its ability to achieve multidimensional imaging without the need for mechanically rotating antennas. By utilizing multiple antennas in a static configuration, MIMO radar systems can capture spatial information along multiple dimensions simultaneously, leading to enhanced imaging capabilities at

reduced cost and complexity. Additionally, MIMO radar systems offer the potential for higher frame rates compared to traditional radar systems. With multiple antennas operating in parallel, MIMO radar can sample the scene more frequently, enabling rapid data acquisition and real-time imaging of dynamic environments. The next step in the advancement of MIMO imaging radars involves the development and optimization of sophisticated signal processing algorithms tailored to exploit the full potential of these systems. These algorithms aim to extract rich spatial information from the received signals, enabling high-fidelity imaging of complex scenes with unprecedented detail and accuracy.

1.3 Imaging Algorithms for 3D MIMO Radar Sensors

Current 3D imaging algorithms employed for many FMCW MIMO radars rely on the Fast Fourier Transform (FFT) technique for generating three-dimensional reconstructions of the observed scene. This algorithm computes each dimension of the image by performing an FFT on a different dimension of the Intermediate Frequency (IF) signal obtained from the radar's input data.

One of the primary strengths of this approach lies in its fast implementation, allowing for rapid processing of radar data. However, despite its efficiency, the FFT-based algorithm suffers from several limitations.

These include near-field distortion issues, restrictions in applicability to Uniform Linear Arrays (ULA), the necessity for array calibration, and the inability to utilize measured antenna gains effectively. Moreover, the algorithm is prone to ringing artifacts, which can degrade the quality of reconstructed images, particularly in complex or cluttered environments.

In response to the limitations of the current FFT-based approach, a novel back projection algorithm is proposed for 3D imaging in the FMCW MIMO radar system. Unlike the FFT-based method, the proposed algorithm offers increased flexibility and versatility in image reconstruction. One of its key advantages is the elimination of ringing artifacts, which commonly plague FFT-based reconstructions. Additionally, the back projection algorithm is capable of addressing near-field distortion issues and is suitable for both near- and far-field imaging scenarios. Moreover, it exhibits compatibility with non-ULA arrays, enabling broader applicability across different radar configurations. A notable feature of the proposed algorithm is its ability to incorporate measured antenna gains effectively, leading to more accurate and reliable imaging results. By leveraging these advantages, the back projection algorithm promises to significantly enhance the imaging capabilities of the FMCW MIMO radar, paving the way for improved performance and expanded applications in various domains.

1.4 Thesis Outline

In this thesis, we embark on a comprehensive exploration of advanced imaging techniques for FMCW MIMO radar systems, aiming to address existing challenges and enhance imaging performance.

Chapter 1 provides a thorough examination of the theoretical background underpinning radar imaging, including signal models and a detailed description of the image reconstruction methods. We delve into the intricacies of both FFT-based and proposed back projection algorithms. Furthermore, a detailed hardware description of the FMCW MIMO radar system under investigation is presented, laying the foundation for subsequent experimental chapters.

Chapter 2 focuses on measurements and validation of hardware stability through extensive static measurements. Through rigorous analysis, we assess the stability of the radar system and measure antenna gains, crucial for accurate imaging.

Chapter 3 shifts the focus to imaging algorithms, detailing the implementation of the back projection algorithm using PyTorch. Additionally, we discuss the interpolation of measured gains and evaluate the performance enhancements achieved through these refinements.

By presenting comprehensive results and analyses, this thesis aims to contribute to the advancement of FMCW MIMO radar imaging technology, with implications for various applications in fields such as remote sensing, autonomous vehicles, and surveillance.

Chapter 2

Radar Fundamentals: Theoretical Background

This section provides a foundation for understanding the principles and methodologies underpinning FMCW MIMO radar imaging. We begin by elucidating the signal model employed in FMCW radar systems, elucidating the distinctions between single-channel and MIMO configurations. Subsequently, we delve into the intricacies of image reconstruction techniques, exploring both the Discrete Fourier Transform (DFT) approach and the proposed backprojection algorithm.

- Window Functions

2.1 Antenna Fundamentals

2.1.1 Physical Background

asdfasdfsdf

2.1.2 Antenna Characteristics

asdfasdf

2.2 Signal Model

Understanding the signal model is fundamental to grasp the operation and capabilities of FMCW MIMO radar systems. In this section, we delve into the intricacies of the signal model, beginning with an exploration of Single Channel Frequency Modulated Continuous Wave (FMCW) radar systems. We clarify the key principles governing their signal generation and processing, laying the groundwork for a comprehensive understanding of radar imaging techniques. Then, we extend our analysis to encompass Multiple Input Multiple Output

(MIMO) FMCW radar systems, highlighting the unique characteristics and complexities associated with their signal model.

2.2.1 Single Channel FMCW

A single channel consists of a transmit antenna and a receive antenna. The transmit antenna sends a so-called chirp of duration T_{chirp} , which is a sinusoid with linearly increasing frequency. The signal $x_{TX}(t)$ send by the transmit antenna is reflected by an ideal point scatterer at position \vec{r}_S and then received at the receive antenna as $x_{RX}(t)$. The propagation delay τ can be calculated using the speed of light c_0 , and the locations of the receive and transmit antennas \vec{r}_{RX} and \vec{r}_{TX} :

$$\tau = \frac{\|\vec{r}_{TX} - \vec{r}_S\| + \|\vec{r}_{RX} - \vec{r}_S\|}{c_0} \quad (2.1)$$

Using a complex representation for the in-phase and quadrature components of the signal, the transmit and receive signal can be formulated for $t \in [0, T_{chirp}]$:

$$x_{TX}(t) = A_0 e^{j(\omega_0 t + \frac{1}{2} \dot{\omega} t^2 + \phi_0)} \quad (2.2)$$

$$x_{RX}(t) = A(\vec{r}_S) x_{TX}(t - \tau) \quad (2.3)$$

$$(2.4)$$

The received signal is then mixed with a copy of the transmitted signal (2.2) and a low-pass filter is applied. The resulting signal $y(t)$ is called *intermittent frequency* signal.

$$y(t) = \text{LP} \{x_{RX}(t) \cdot x_{TX}(t)\} \quad (2.5)$$

$$= \text{LP} \left\{ A_0 e^{j(\omega_0 t + \frac{1}{2} \dot{\omega} t^2)} \cdot A(\vec{r}_S) A_0 e^{j(\omega_0(t-\tau) + \frac{1}{2} \dot{\omega}(t-\tau)^2)} \right\} \quad (2.6)$$

$$= A_0^2 A(\vec{r}_S) e^{j(\frac{1}{2} \dot{\omega} \tau^2(\vec{r}_S) - \omega_0 \tau)} \cdot \text{LP} \left\{ e^{j(2\omega_0 t + \frac{1}{2} \dot{\omega} t^2 - \dot{\omega} \tau t)} \right\} \quad (2.7)$$

$$\approx G(\vec{r}_S) e^{-j\dot{\omega} \tau t} \quad (2.8)$$

The fact that the IF-signal contains all the information – i.e. the IF signal's frequency directly corresponds to the target's distance – explains the main advantage of this technology. The carrier frequency can be orders of magnitude higher than the intermittent frequency, which drastically reduces the requirements for the subsequent signal processing, while retaining the improved resolution due to the smaller wavelengths of the carrier frequency.

FIND QUOTE: "GHz resolution for MHz processing"

To locate a target in the cross-range dimensions, a single-channel FMCW-radar can be used to scan in multiple directions, by either rotating the antennas, redirecting their beam with rotating mirrors, or with beamforming antenna arrays. In any case, this requires highly directive antennas and also increases size, weight and cost of a radar sensor.

2.2.2 Uniform Linear Arrays

2.2.3 Virtual Arrays

2.2.4 Multiplexing Techniques

Multiple-input multiple-output radar benefits from increased diversity and signal power. If N_{TX} transmit antennas and N_{RX} receive antennas are employed, $K = N_{TX} \cdot N_{RX}$ channels are available. To differentiate the signals from each other, a multiplexing technique has to be chosen. Options include time division multiplex, frequency division multiplex and code division multiplex.

In TDM, multiple access is achieved by the transmit antennas all send one after another, while all receive antennas receive simultaneously. In FDM, simultaneous transmission is made possible by subdividing the bandwidth and assigning a different frequency range to each antenna. That means that TDM allows for higher bandwidths for each transmission, while FDM allows higher transmission durations.

In CDM, both simultaneous transmission and use of the entire bandwidth is made possible by using a different waveform to each channel. However, processing at the carrier frequency is required to differentiate the signals from another, as opposed to TDM and FMD, where all processing can be done at the intermittent frequency range.

Depending on the application, a compromise has to be found between the advantages and drawbacks of each method. There are also methods available that combine aspects of these three basic paradigms, such as OFDM and Hadamard-Coding.[citation needed]

Once the received signals are demultiplexed, the ideal receive signal for antenna pair $k \in \{0, 1, \dots, K - 1\}$:

$$y_k(t) = G_k(\vec{r}_S) e^{-j\omega\tau_k(\vec{r}_S)t} \quad (2.9)$$

Note that both the gain and the propagation delay may differ from channel to channel.

In reality, the scene can consist of multiple and expansive scatterers, that reflect the transmitted signals at different intensities. which is summarized as a locational reflectivity $F_k(\vec{r})$.¹ Also, interference and electric noise may be present in each channel, which we summarize as $n_k(t)$. Thus, the overall IF-signal is:

$$y_k(t) = \iiint F_k(\vec{r}) G_k(\vec{r}) e^{-j\omega\tau_k(\vec{r})t} d\vec{r} + n_k(t) \quad (2.10)$$

$$(2.11)$$

¹The index k is introduced here to take obstructed visibility into account: from the point of view of one channel, two scatterers may be visible simultaneously, while from the point of view of another, one might obstruct the other's visibility.

After sampling the signal at sampling intervals T_s such the sampling frequency $f_s = \frac{1}{T_s}$ is sufficiently high: $2f_s > \frac{1}{2\pi}(\omega_0 + \dot{\omega}T_{chirp})$, and with M samples such that $MT_s < T_{chirp}$, the sampled IF-signal can be defined as:

$$y_k[m] = y_k(t = mT_s), \text{ for } m \in \{0, 1, \dots, M-1\} \quad (2.12)$$

2.3 Image Reconstruction

Image reconstruction is an inverse problem where the locational reflectivity of the scene $F(\vec{r})$ has to be estimated from the received signals $y_k[m]$. Multiple approaches are available; in the following, three will be presented.

2.3.1 Discrete Fourier Transform

The discrete fourier transform can be implemented with high efficiency, and many CPUs even include silicone-based implementations [citation needed]. In this approach, the DFT is applied over three dimensions of the input signal, obtaining a discrete output signal in spherical coordinates whose amplitude is an estimate of the locational reflectivity.

For each input channel, the range of a target can be estimated by applying the DFT over time. The resulting spectrum's peak corresponds to the target:

$$\mathcal{F}_m\{y_k[m]\}(\Omega) = \sum_{m=0}^{M-1} e^{-j2\pi\frac{m\Omega}{M}} y_k[m] \quad (2.13)$$

$$= G_k(\vec{r}_S) \delta(\Omega - \dot{\omega} \tau_k(\vec{r}_S) T_s) \quad (2.14)$$

In order to understand how information on the direction of a target can be extracted from the channel data, we consider an ideal $1 \times K$ horizontal uniform linear array (*ULA*) where the spacing is exactly $d = \frac{\lambda_0}{2}$, with $\lambda_0 = \frac{c_0}{f_0}$. The antennas are located at $\vec{r}_{TX} = \vec{0}$ and $\vec{r}_{RX,k} = (kd, 0, 0)^T$. A scatterer located at $\vec{r}_S = (r_S \sin \theta_S, r_S \cos \theta_S, 0)^T$ reflects the transmitted radar waves with an intensity of A_S . Then, their runtime across the array is:

$$\tau_k = \frac{1}{c_0} (\|\vec{r}_{TX} - \vec{r}_S\| + \|\vec{r}_{RX,k} - \vec{r}_S\|) \quad (2.15)$$

$$(2.16)$$

In far-field conditions, the target is far enough ($r \gg Kd$) away for the reflected wavefronts to be planar. That means that the runtime can then be approximated as such:

$$\tau_k \approx 2r_S + kd \sin \theta_S \quad (2.17)$$

In equation 2.7 it can be seen that the locational gain $G(\vec{r})$ contains a phase shift depending on the runtime of the waves:

$$G_k(\vec{r}) = A_S A_k(\vec{r}) e^{j(\frac{1}{2}\dot{\omega}\tau_k^2 - \omega_0\tau_k)} \quad (2.18)$$

$$\approx A_S A_k(\vec{r}) e^{-j\omega_0\tau_k} \quad (2.19)$$

Assuming the attenuation along the path is channel-independent, the signals can all be considered copies of each other:

$$y_k(t) = A_S A_k(\vec{r}) e^{-j\omega_0\tau_k} e^{-j\dot{\omega}\tau_k t} \quad (2.20)$$

$$= A_S A_k(\vec{r}) e^{-j\frac{\omega_0}{c_0}(2r_S + kd \sin \theta_S)} e^{-j\frac{\dot{\omega}}{c_0}(2r_S + \overbrace{kd \sin \theta_S}^{\ll 2r_S})t} \quad (2.21)$$

$$= A_S A_0(\vec{r}) e^{-j\frac{\omega_0}{c_0}2r_S} e^{-j\frac{\dot{\omega}}{c_0}2r_S t} e^{-j\frac{\omega_0}{c_0}kd \sin \theta_S} \quad (2.22)$$

$$= y_0(t) e^{-j\frac{\omega_0}{c_0}kd \sin \theta_S} \quad (2.23)$$

$$\Rightarrow y_k[m] = y_k(t = mT_S) = y_0[m] e^{-j\frac{\omega_0}{c_0}kd \sin \theta_S} \quad (2.24)$$

Applying the DFT accross the ULA yields:

$$\mathcal{F}_k\{y_k[m]\}(\Omega) = \mathcal{F}_k\{y_0[m] e^{-j\frac{\omega_0}{c_0}kd \sin \theta_S}\}(\Omega) \quad (2.25)$$

$$= y_0[m] \cdot \delta\left(\Omega - \frac{\omega_0}{c_0}(d \sin \theta_S)\right) \quad (2.26)$$

$$= y_0[m] \cdot \delta(\Omega - \pi \sin \theta_S) \quad (2.27)$$

$$(2.28)$$

The azimuth angle θ_S can be extracted from the signal supplied by a horizontal ULA. Analogously, the elevation angle ϕ_S can be obtained with a vertical ULA. If a $1 \times K$ array is used, where the K receive antennas form a uniformly spaced grid, successive DFTs across the rows and columns of this grid yield two dimensions. However, the same can be achieved with fewer antennas in a MIMO configuration.

The *virtual array* of a MIMO array is a corresponding SIMO array. It has the same number of channels: if the original array is $N_{TX} \times N_{RX}$, the virtual array has $1 \times K$ channels, where $K = N_{TX} \cdot N_{RX}$.

The virtual transmit and receive antenna associated with channel $k = 0$ are placed in the origin of the virtual array's coordinate system. The other receive antennas are then placed such that the displacement between them and the transmit antenna is the same as it was between the original MIMO array's corresponds transmit and receive antenna. It is possible for virtual antennas to be at the same location.

Overall, a 3D image in range, azimuth, and elevation is generated by calculating the DFT over time, and the DFTs over the rows and columns of the

virtual array. For this to work, the scatterer needs to be distant enough for the wavefronts to be planar, and the virtual array's grid needs to be uniformly spaced with $d = \lambda_0/2$ spacing.

2.3.2 Backprojection

Compared to the DFT-based approach, backprojection takes fewer approximations and requirements on the array to work, while using a similar amount of computation. The approach works by correlating the input signal $y_k[m]$ to the theoretical signal $s_k[m, \vec{r}]$ of an ideal scatterer at different locations. The mean correlation of all channels to the theoretical signal is then used as an estimate for the locational reflectivity:

$$\hat{F}(\vec{r}) = \frac{1}{K} \sum_{k=0}^{K-1} s_k[m, \vec{r}] \star y_k[m] \quad (2.29)$$

$$= \frac{1}{K} \sum_{k=0}^{K-1} \sum_{m=0}^M s_k^*[m, \vec{r}] y_k[m] \quad (2.30)$$

Using the signal model from (2.9) yields:

$$\hat{F}(\vec{r}) = \frac{1}{K} \sum_{k=0}^{K-1} \sum_{m=0}^M G_k^*(\vec{r}) e^{+j\dot{\omega}\tau_k(\vec{r}_S)mT_s} y_k[m] \quad (2.31)$$

To reduce the computational intensity of this algorithm, calculating the inner sum (over m) can be rewritten as an inverse discrete fourier transform (IDFT):

$$\hat{F}(\vec{r}) = \frac{1}{K} \sum_{k=0}^{K-1} G_k^*(\vec{r}) \sum_{m=0}^M e^{+j\dot{\omega}\tau_k(\vec{r}_S)mT_s} y_k[m] \quad (2.32)$$

$$= \frac{1}{K} \sum_{k=0}^{K-1} G_k^*(\vec{r}) \sum_{m=0}^M e^{j\Omega m} y_k[m] \Big|_{\Omega=\dot{\omega}\tau_k(\vec{r}_S)T_s} \quad (2.33)$$

$$= \frac{1}{K} \sum_{k=0}^{K-1} G_k^*(\vec{r}) \mathcal{F}_m^{-1} \{y_k[m]\} (\Omega = \dot{\omega}\tau_k(\vec{r}_S)T_s) \quad (2.34)$$

2.3.3 MUSIC

The Multiple Signal Classification (MUSIC) algorithm can also be used to estimate the locational reflectivity of a scene. It operates on the time-domain fourier transform of the IF-signal, and makes similar far-field approximations as the DFT-based approach. The abstract signal model for MUSIC is:

$$\mathbf{y}(t) = \mathbf{A} \cdot \mathbf{s}(t) + \mathbf{n}(t) \quad (2.35)$$

Here, $\mathbf{y}, \mathbf{n} \in \mathbb{C}^K$, $\mathbf{A} \in \mathbb{C}^{K \times Z}$, and $\mathbf{s} \in \mathbb{C}^Z$. Z is the number of voxels in the output image and K the number of receive channels. For example, if the output image consist of $X \times Y \times Z$ cuboid voxels, then $Z = X \cdot Y \cdot Z$.

Thus, the support matrix \mathbf{A} is a linear transform from the locational reflectivity \mathbf{s} to the expected input signal vector \mathbf{y} . Each collumn vector \mathbf{a}_z of the support matrix \mathbf{A} therefor corresponds to the expected input signal vector caused by a point source.

The MUSIC algorithm revolves around the correlation matrix of its input signal $\mathbf{R}_{\mathbf{yy}}$. Assuming the a stationary scene with zero-mean noise of covariance $\mathbf{C}_{\mathbf{nn}}$, it follows that

$$\mathbf{R}_{\mathbf{yy}} = \mathbb{E}\{\mathbf{y}\mathbf{y}^H\} \quad (2.36)$$

$$= \mathbf{A}\mathbf{R}_{\mathbf{ss}}\mathbf{A}^H + \mathbf{C}_{\mathbf{nn}}, \text{ with } \mathbf{R}_{\mathbf{ss}} := \mathbb{E}\{\mathbf{s}\mathbf{s}^H\} \quad (2.37)$$

Assume that $\mathbf{R}_{\mathbf{ss}}$ is nonsingular with rank q and that \mathbf{A} has full rank. If $\mathbf{R}_{\mathbf{yy}}$ has p eigenvalues, then the smallest $p - q$ of them are all σ^2 , and their corresponding eigenvectors – i.e. the collumns of $\mathbf{C}_{\mathbf{nn}}$ – are all orthogonal to the support vectors \mathbf{a}_z .

This property is key to the MUSIC algorithm. The metric used to generate an image is the projection of \mathbf{a}_z onto the $\mathbf{C}_{\mathbf{nn}}$. Due to their orthogonality, the projection of support vectors corresponding to a signal source will be zero. The image intensity at voxel z is thus defined computed as the normalized inverse square magnitude of this projection:

$$P_{MUSIC}[z] = \frac{\mathbf{a}_z^H \mathbf{a}_z}{\mathbf{a}_z^H \hat{\mathbf{C}}_{nn}^H \hat{\mathbf{C}}_{nn} \mathbf{a}_z} \quad (2.38)$$

The input signals are often highly correlated, due to phenomena such as multipath propagation or inter-channel crosstalk. This unfortunately means that nonsingularity of $\mathbf{R}_{\mathbf{ss}}$ cannot always be guaranteed. A preprocessing step is required to “decorrelate” the signals and thereby making $\mathbf{R}_{\mathbf{ss}}$ singular again.

While early schemes, such as the “3/4in plywood” spacial dither algorithm by Widrow *et al.* [CITE] consisted of mechanically moving the receive antenna array orthogonal to the look direction, preprocessing can also be done after receiving the signal.

Spacial smoothing, as proposed by [CITE], improves the correlation matrix’s eigenstructure by

Chapter 3

Hardware Evaluation

This chapter starts with a brief introduction of the sensor employed in this thesis(sec. 3.1). The system properties and specifications are outlined, and the makeup of the antenna array is described. On the basis of this knowledge, the system can be evaluated through a number of experiments.

The second part of this chapter (sec. 3.2) focuses on stability analysis, which is conducted through long-term static measurements of a corner reflector. By monitoring the radar system's performance over an extended period, we aim to assess its stability and reliability in real-world operating conditions. The analysis provides insights into any temporal variations or drifts in system parameters, enabling proactive measures to mitigate potential sources of error.

The third part of this chapter (sec. 3.3) focuses on antenna gain measurements using a rotating setup. Antenna gain plays a crucial role in radar imaging, affecting the system's sensitivity and resolution. By rotating the radar system and precisely measuring the received signals from known targets, we can accurately determine the antenna gain across different azimuth angles. This measurement process enables the characterization and validation of antenna performance, facilitating improved imaging accuracy and consistency.

Through these procedures, Chapter 3 aims to establish a robust foundation for the subsequent imaging algorithms that are implemented and evaluated in Chapter 4. By ensuring the stability of system parameters and accurately characterizing antenna performance, we strive to enhance the reliability and effectiveness of FMCW MIMO radar imaging for various applications.

3.1 Indurad Multi-Channel Radar

The sensor employed in this thesis operates on Multiple Input Multiple Output (MIMO) Frequency Modulated Continuous Wave (FMCW) technology, showcasing advanced features tailored for precise radar imaging.

- applications

3.1.1 Sytem Parameters

It offers a range capability of less than 100 meters, extendable up to 800 meters with active beamforming techniques. The sensor achieves a remarkable range resolution of 3.8 millimeters, enabling detailed imaging of objects within its detection range.

Utilizing a sawtooth or chirp signal type, the sensor employs Time Division Multiplexing (TDM) in Transmission (Tx) for efficient multiplexing. Operating within the frequency range of 77 to 81 gigahertz (GHz), it leverages the millimeter-wave spectrum to achieve high-resolution imaging suitable for a variety of applications.

The chirp duration of the sensor is between 60 to 70 microseconds, ensuring effective signal processing and data acquisition. Equipped with 12 transmit (Tx) antennas and 16 receive (Rx) antennas, the sensor offers comprehensive coverage and sensitivity, facilitating robust imaging performance.

Furthermore, the sensor's Intermediate Frequency (IF) samplerate is set at 22 megahertz (MHz), providing sufficient bandwidth for accurate signal processing and analysis. This combination of advanced features and specifications positions the sensor as a versatile and effective tool for radar imaging tasks in diverse scenarios, ranging from automotive safety systems to industrial sensing applications.

- Summarize in tables
- Image of the sensor
- HW: Cascaded TI-AWR2243P, FPGA, embedded linux, processing

3.1.2 Virtual Antenna Array

"As described in section ..., a virtual antenna array ..."

- Antenna Array Drawing
- Virtual Antenna Array Drawing
- Table of measurements

3.2 Stability Analysis

In this section, we delve into evaluating the stability of the radar system through a comprehensive analysis of its response under static conditions. By observing the system’s behavior over an extended period, our objective is to identify and characterize any temporal variations or drifts in system parameters. This assessment is crucial for understanding the inherent stability of the radar system and for identifying potential sources of instability that may impact imaging performance. Through rigorous stability analysis, we aim to enhance the accuracy, consistency, and reliability of radar imaging results, laying a solid foundation for subsequent experiments and applications.

In our stability analysis, we meticulously investigate several key parameters to assess their impact on radar system performance and calibration. These parameters include system restart, time elapsed since system restart, system temperature, and the frequency of the deramped signal. Each of these parameters plays a crucial role in determining the stability and reliability of radar imaging, making them essential targets for investigation during calibration procedures.

Firstly, we examine the influence of time elapsed since system restart on radar stability. Over time, certain components of the radar system may undergo gradual changes or drifts, affecting signal quality and imaging precision. By monitoring the system’s response over varying time intervals, we can assess the magnitude and nature of these temporal variations and devise strategies to mitigate their impact on calibration.

Secondly, we scrutinize the effects of system restarts on radar performance. System restarts can introduce transient variations in system behavior, potentially leading to fluctuations in signal characteristics and imaging accuracy. Understanding how the radar system responds to restart events is critical for ensuring consistent performance and minimizing calibration errors.

Furthermore, we investigate the dependence of radar system performance on system temperature. Temperature fluctuations can influence the characteristics of electronic components, potentially introducing biases and inaccuracies in radar measurements. By quantifying the effects of temperature variations on radar stability, we can implement temperature compensation techniques to enhance calibration accuracy and reliability.

3.2.1 Setup

The sensor is placed in a low-reflection environment and a corner reflector is placed at boresight in front of the sensor at a distance of roughly 1.60 m (cf. 3.1). The time data collected in all channels is recorded every minute. The temperature readings of the sensor’s CPU, FPGA and radar frontend are also recorded every minute (cf. 3.2). The experiment is repeated with a different ramp slope $\dot{\omega}$ in order to achieve a different frequency of the deramped signal, since

$$f_{IF} = \dot{\omega} \cdot \frac{2r_{refl}}{c} \quad (3.1)$$

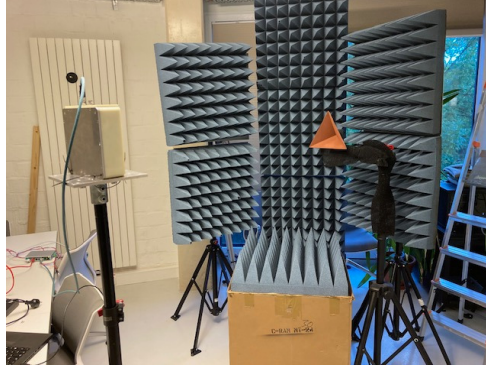


Figure 3.1: Measurement Setup

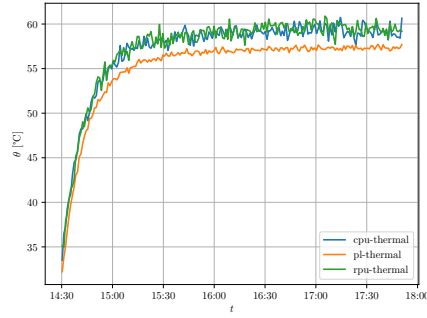


Figure 3.2: Measured System Temperature After Startup

Due to the geometry of the setup, the runtime of each transmitted wavefront should be identical. Thus, the ideal deramped signal should be of a single, constant frequency and without inter-channel phase differences. In practice, the reflector peak will have a certain bandwidth, and other peaks at higher and lower frequencies will be present due to incomplete shielding and/or unwanted reflections. Also, the reflector peak may wander if the setup geometry moves. For analysis, system runtime and temperature cannot be considered independent variables, as illustrated in figure 3.3: The system starts at ambient temperature, heating up and approaching a stable operating temperature on turning on (t_0), and cooling back down after turning off (t_1).

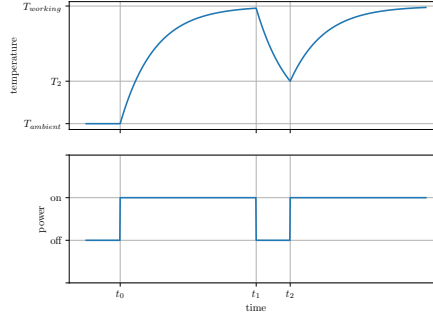


Figure 3.3: Expected System Temperature over Time for Interrupted Power

3.2.2 Preprocessing and Analysis

Because of the aforementioned imperfections in the experiment, some preprocessing is required to analyze the systematic offsets present in the radar signal. Multiple additional peaks in the spectrum are visible in figure 3.4; indeed, the maximum peak is not even caused by the reflector. It is therefore necessary to limit the analysis to only the FFT-bin at the maximum of the peak caused by the reflector.

The measurement environment also exhibits some minor changes in temperature and humidity, which can result in the geometry to shift by a few millimeters. This shows by the spectral maximum shifting over time, as seen in figure 3.5.

In the following analysis, the main metric is the complex ratio of the maximum FFT-bin to its initial value. Using a logarithmic representation of this ratio, it can be represented as a level difference [dB] and a phase difference [°].

3.2.3 Effects of System Temperature and Runtime

Multiple measurements have indicated that, while the amplitude rarely varies by more than 1 dB, the phase is not as stable over time. Typically, the mean phase drifts by up to 50° in the hours after system startup, with the rate of change reducing after around four hours. However, it has to be noted that there is no clear correlation between phase drift and temperature: the phase continues to drift after the system temperature stabilizes; the reduction in drift only occurs hours after the system has reached a stable temperature of approximately 60 °C.

TODO : split new measurement into heating up (23:00-8:00) and stable (13:00-12:00).

3.2.4 Effects of Self-Calibration

As described in the specification, the AWR2243P-chips undergo a self-calibration upon initialization. This initialization can be triggered by either restarting the

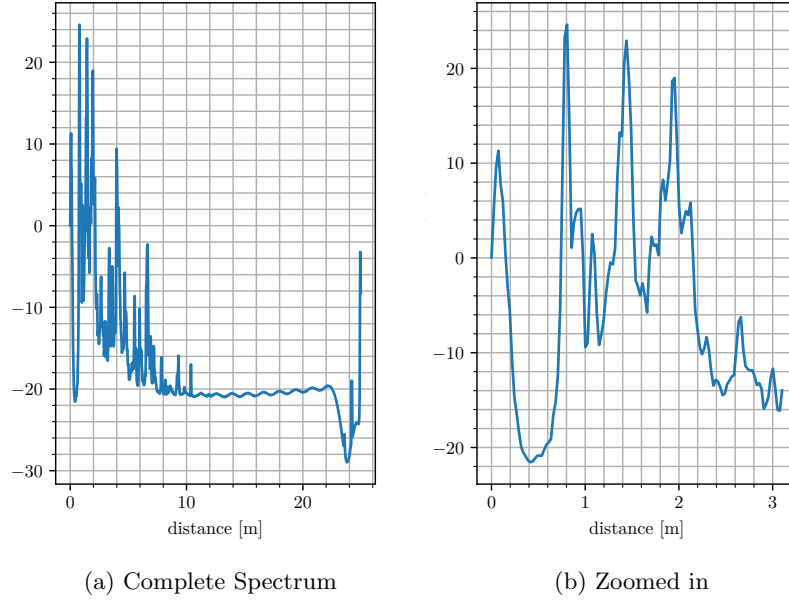


Figure 3.4: Mean Intensity Spectrum

entire system or by re-writing the configuration registers on the radar chips. Indeed, this initial calibration can be observed in the data (cf. 3.7). After the connection with the radar has been re-established, the following effects are visible:

- slightly increased level: the level in each channel increases by approximately 0.3 dB
- increased incoherence: after the first restart of the system, the drift in both level and phase is distributed more broadly
- mean phase: the mean phase drift changes by up to 5°

TODO: graph jumps to new phase before reboot. confirm or fix.

TODO: quantify amount of discontinuity

3.2.5 Antenna Separation

TODO: split gains like in 3.3.7

3.2.6 Differences between Antennas

TODO: Plots, antenna Separation

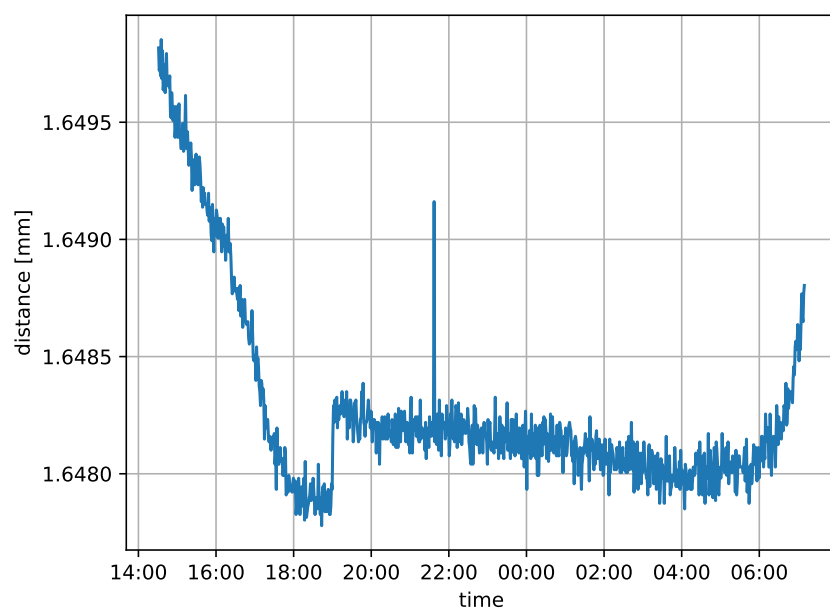


Figure 3.5: Change in reflector distance as measured by iMCR

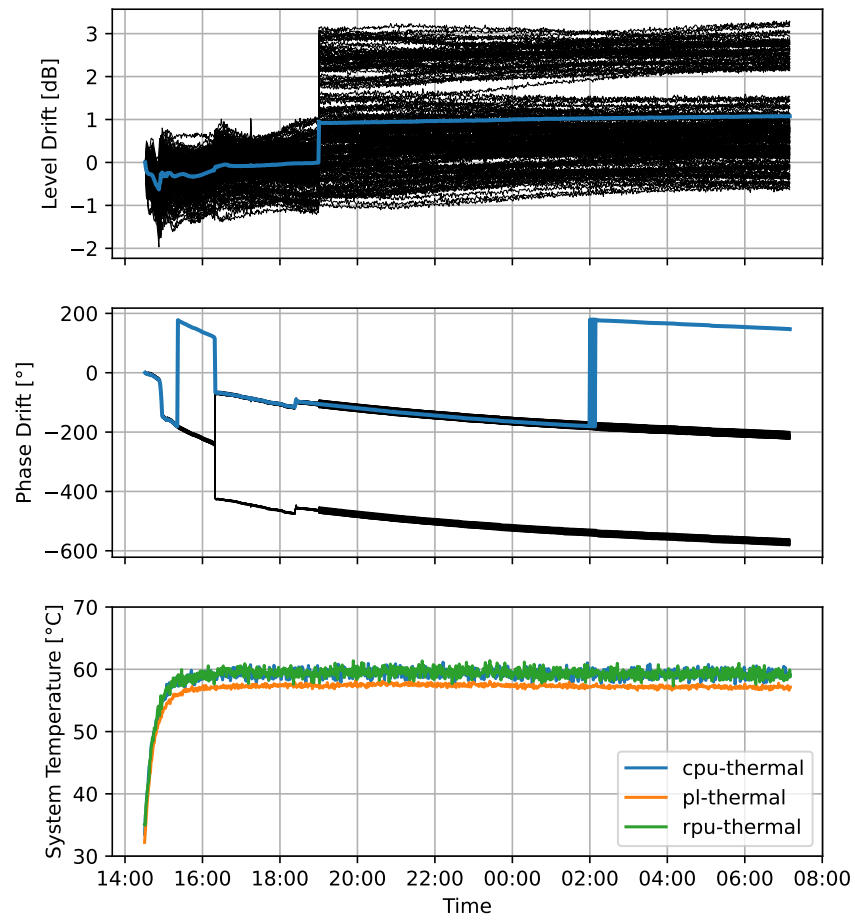


Figure 3.6: Recorded drift over night

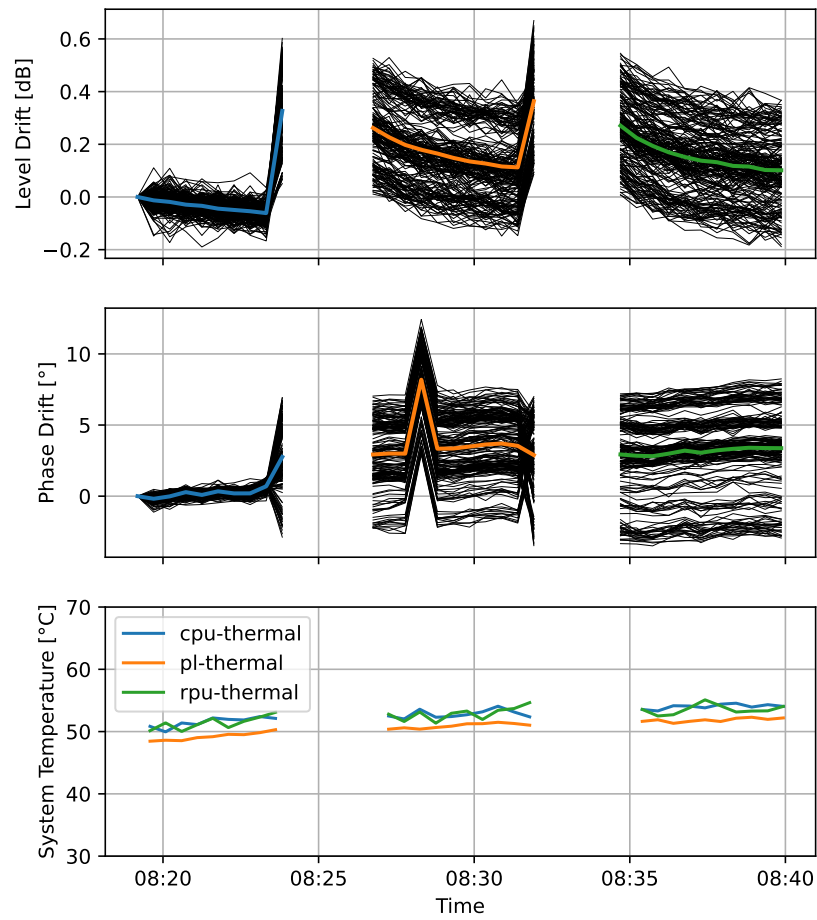


Figure 3.7: Recorded drift and temperature with system restart

Comparing the amplitude recorded by all antenna pairs, no channel in particular stands out [PLOT NEEDED]. When looking at the different channels phase drifts, it can be seen that the array becomes less coherent over time. The distribution of phase shift across channels seems roughly gaussian [PLOT NEEDED], with increasing variance over time. No antenna pair seems more prone to incoherence than the rest: across measurements, the maximum outlier (in terms of divergence from the mean phase) cannot be associated with a chip or antenna.

3.2.7 Conclusion

The systematic offsets of the system are now better described. While the amplitude of the measured signal is relatively stable, the phase is affected more strongly. All channels' phases drift from both their initial value and each other over time. It can also be demonstrated that restarting the radar frontend has an effect on the phase, since the frontend undergoes an automatic calibration each time. No clear bias within the array has been found, seeing that across multiple measurements, all channels seem to be affected similarly.

Overall, it has been demonstrated that the radar system cannot be considered ideal and its imaging fidelity will be affected by the growing incoherence during long runtimes. Furthermore, restarting the system will change the systematic offsets due to the automatic calibration of the radar frontend hardware. Adaptive online calibration would be required to deal with these effects.

To limit the scope of this thesis, the focus is set on the offline-part of the calibration process. Because the phase and amplitude offsets are affected by the automatic calibration of the system, this calibration should ideally be repeated on every startup, ideally letting the system temperature stabilize.

An advantage for the offline calibration is that the speed at which the array incoherence grows is limited. That means that for sufficiently short runtimes, it can be assumed to be static, and therefor, an offline calibration to be sufficient.

3.3 Antenna Characteristics

In this section, the array’s antenna gains are measured and then analyzed. The measured antenna gains can then directly be employed as weights for the back-projection algorithm, resulting in a calibrated array. A subset of the extrated paramters can also be employed for offline calibration when using FFT-based imaging

3.3.1 Setup

TODO: include photo

To measure the antenna gains locationally, the sensor is mounted on a vertical rotating axis. Rotating the axis allows a corner reflector to be observed from multiple angles, while retaining a fixed distance. Doing this measures the antenna gain in azimuth; to measure the antenna gain in elevation, the sensor is mounted on the same axis to rotate around its horizontal axis. These two measurements are repeated for multiple reflector distances.

While the relative position of the rotating axis can be recorded with a decent resolution of around 0.01° , the absolute positioning of the reflector is hard to achieve with the same accuracy. The sensor itself can however be used to improve positioning: during setup, the phase of the peak caused by the reflector is displayed. Mimimizing the phase differences between the array’s channels ensures that the reflector be located at boresight. These measurements are repeated with multiple distances of the relector.

The measurements are conducted outdoors to make room for longer distance measurements. This however means that the recorded scene is not free of interference.

Several trees and brushes are located at the edge of the scene. Extra interference and reflections are caused by the floor. It is also not perfectly level, varying in height by up to 1 m, which means that the reflector is not always exactly at the same height. The sensor mount also causes some problems [NEEDS DRAWING]: when the sensor is mounted on its inside, the rotational axis passes exactly through the antenna plane, but some shading is occurs when the reflector is on the same side as the sensor mount’s diagonal reinforcement. When the sensor is mounted on the outside, the rotational axis is no longer in the antenna plane, adding complexity and inaccuracy to the subsequent analysis.

3.3.2 Preprocessing and Analysis

TODO: Redo Signal Model with Antenna Gains and k as multi-index

$$G(r, \theta, \phi) = \frac{1}{r^2} C e^{j\varphi} c(\theta, \phi)$$

$$C e^{j\varphi} = A e^{j\psi} \cdot B e^{j\vartheta}$$

The expected behavior is formulated in the signal model (see equation 2.8ff.):

$$y_k(t = mT_s) = G_k(\vec{r}) e^{-j\dot{\omega}\tau_k mT_s}, \quad (3.2)$$

$$\text{where } G_k(\vec{r}) = \frac{A_k A_S(\theta_S, \phi_S)}{R_k^2} e^{-j\omega_0 \tau_k + \varphi_k} \quad (3.3)$$

$$\text{and } A_k := A_{Tx,i} \cdot A_{Rx,j} \text{ for } k = N_{Tx}i + j \quad (3.4)$$

A constant channel gain A_k , resulting from the transmitting and receiving antenna gains, is attenuated with increasing distance, with $|G_k| \propto \frac{1}{R_k^2}$. The effect of the angle of arrival on the gain is represented by the factor $A_S(\theta_S, \phi_S)$, which is normalized such that $A_S(\theta_S, \phi_S) \in [0, 1]$. A constant channel phase offset φ_k is set back proportionally to the time of flight $\tau_k = 2R_k/c$.

The goal of the subsequent analysis is twofold: to extract all parameters of the above signal model while identifying and removing interference through processing.

Limiting the analysis to a single range bin will work towards both reducing interference and extracting parameters: ideally, the DFT spectrum of the IF signal consists of just a single peak, weighted with the locational gain $G_k(\vec{r})$ (see eqn. 2.14). Thus, the parameters of the locational gain can be extracted if the correct bin $\hat{\Omega}$ is picked:

$$\hat{\Omega} = \dot{\omega}\tau_k(\vec{r}_S)T_s \quad (3.5)$$

$$= \dot{\omega} \frac{2R_k}{c} T_s \quad (3.6)$$

$$\Rightarrow \mathcal{F}_m\{y_k[m]\}(\Omega = \hat{\Omega}) = G_k(\vec{r}_S) \quad (3.7)$$

The relation between a range bin Ω and its corresponding range r is

$$\Omega = r \cdot \frac{4N_{fft}\dot{\omega}}{c_0 f_s} \quad (3.8)$$

3.3.3 Range Estimation

In order to improve the accuracy of the subsequent analysis, the exact position of the reflector is estimated from the radar data. To do this, numerical optimization is employed to find the parameter set $\hat{R}_s, \hat{\theta}_s, \hat{\epsilon}$ that minimizes the

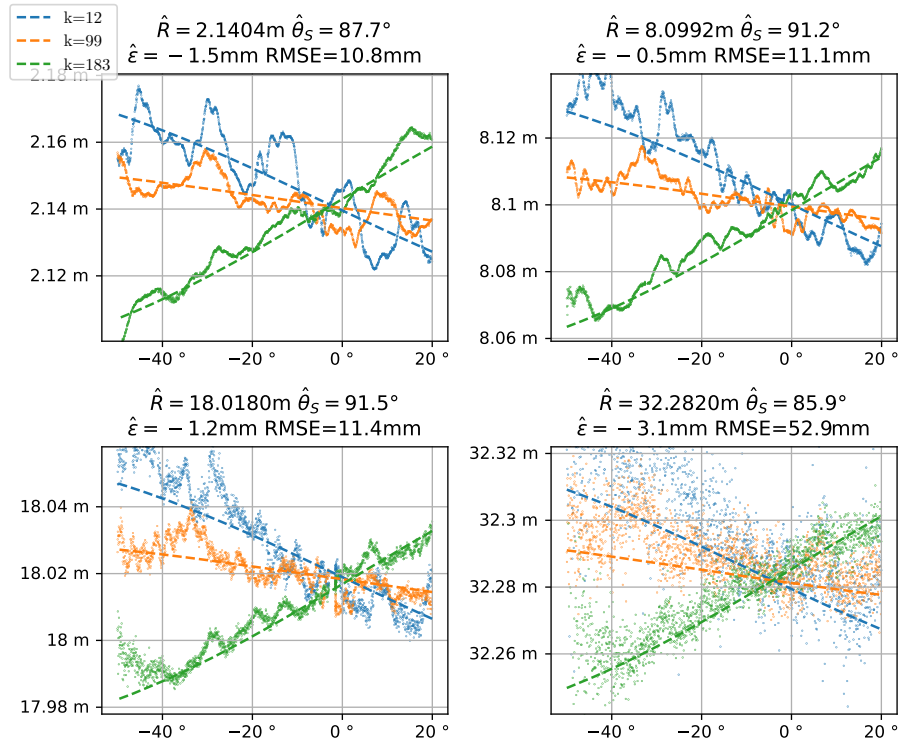


Figure 3.8: Example Optimization Results

difference between a range estimate $\hat{R}_{k,l}$ and the range spectral peak $R_{k,l}$ for all channels k and recorded orientations l :

$$R_{k,l} = \arg \max_r \mathcal{F}_m\{y_{k,l}[m]\} \left(\Omega = \frac{N_{fft}r}{R_{max}} \right) \quad (3.9)$$

The range estimate $\hat{R}_{k,l}$ is determined geometrically from the distances of each channel's transmit and receive antenna to the reflector. If the sensor is mounted to rotate in azimuth, the range depends on the rotation angle θ_l :

$$\hat{R}_{k,l} = \frac{\|\vec{r}_{TX,k} - \vec{r}_S(l)\| + \|\vec{r}_{RX,k} - \vec{r}_S(l)\|}{2} \quad (3.10)$$

$$\vec{r}_{TX,k} = \begin{bmatrix} x_{TX,k} \\ y_{TX,k} \\ \hat{\epsilon} \end{bmatrix}, \vec{r}_{RX,k} = \begin{bmatrix} x_{RX,k} \\ y_{RX,k} \\ \hat{\epsilon} \end{bmatrix}, \quad (3.11)$$

$$\vec{r}_S(l) = \begin{bmatrix} (\hat{R}_S - \hat{\epsilon})\sin(\hat{\theta}_S - \theta_l) \\ 0 \\ (\hat{R}_S - \hat{\epsilon})\cos(\hat{\theta}_S - \theta_l) \end{bmatrix}, \quad (3.12)$$

If it is mounted to rotate in its elevation, the reflector appears to move depending on the rotation in elevation ϕ . For a given rotation angle ϕ_l , it is located at

$$\vec{r}_S(l) = \begin{bmatrix} 0 \\ (\hat{R}_S - \hat{\epsilon})\sin(\hat{\phi}_S - \phi_l) \\ (\hat{R}_S - \hat{\epsilon})\cos(\hat{\phi}_S - \phi_l) \end{bmatrix}, \quad (3.13)$$

In our setup, a signal is recorded with θ (or ϕ , respectively) from 0° to 180° , with the reflector located at roughly $R_S = \{2, 8, 18, 32\}$ m and either $\theta_S = 90^\circ$ or $\phi_S = 90^\circ$.

With this, the loss \mathcal{L} can be defined as the mean square difference between \hat{R} and R_p , the mean being computed as the average over all K channels and L sensor rotations:

$$\mathcal{L}(\hat{R}_s, \hat{\theta}_s, \hat{\epsilon}) = \frac{1}{KL} \sum_{l=0}^{L-1} \sum_{k=0}^{K-1} (R_{k,l} - \hat{R}_{k,l}(\hat{R}_s, \hat{\theta}_s, \hat{\epsilon}))^2 \quad (3.14)$$

Figure 3.8 shows the result of 100 iterations at an adaption rate of 0.05 at the example of three different channels, comparing the measured spectral peaks (shown as points) to the estimated position (dotted line) of the reflector. Only a subset of orientations is used for the estimate, namely $-50^\circ < \theta_l - 90^\circ < 20^\circ$.

For the first three measurements, the peak location oscillates around the estimate by 1 to 2 cm. The root mean square error is reduced to 11 mm. The fourth measurement has higher noise, with no clear line being visible in the spectral peaks.

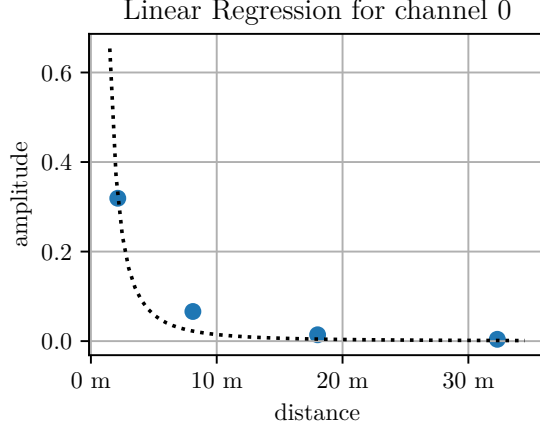


Figure 3.9: Example Channel Gain Linear Regression

3.3.4 Amplitude

First, the recorded amplitudes are analyzed, beginning with the influence of the angle of arrival. To do this, each channel's range FFT spectrum's amplitude is evaluated at the respective calculated peak location $\hat{\Omega}$, and then normalized by dividing it by its maximum.

The first parameters to extract are the three factors A_{Tx} , A_{Rx} and $A_S(\theta_S, \phi_S)$. Since the angle of arrival's impact has been constrained to $[0, 1]$, we know that

$$\max_{\theta, \phi} |G_k(R, \theta, \phi)| = \frac{A_k}{R^2} \quad (3.15)$$

With that, an estimate of the channel gain A_k at exactly 1 m can be obtained via linear regression. An example linear regression is shown in figure 3.9, and the resulting estimates with their standard deviations are summarized in figure 3.10.

Having estimated the channel gains, we can now turn to analysing the angle-of-arrival characteristics. Assuming the optimal range bin $\hat{\Omega}$ is evaluated each time, we get

$$\left| \mathcal{F}_m \{y_k[m]\}(\Omega = \hat{\Omega}) \right| = \frac{A_k A_S(\theta_S, \phi_S)}{R_k^2} = |G_k(\vec{r}_S)| \quad (3.16)$$

Dividing that by the maximum gain (3.15) yields

$$\frac{|G_k(\vec{r}_S)|}{\max_{\theta, \phi} |G_k(R, \theta, \phi)|} = A_S(\theta, \phi) \quad (3.17)$$

Figure 3.11 provides an overview by displaying the mean channel characteristic for each measurement. It can be seen that the gain is strongest when

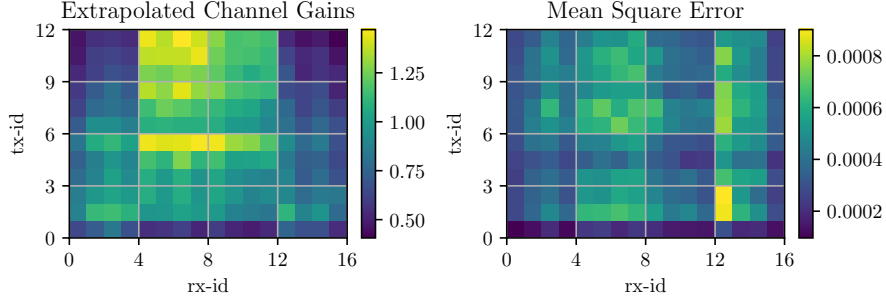


Figure 3.10: Channel Gain Linear Regression Results

the target is directly in boresight, tapering off when the target is off-center. The reduction in gain is stronger when the target moves off to the side in the elevation, than in azimuth. While the target remains stronger than background noise up until an azimuth angle of around -75° to 75° , it can only be seen in an elevation angle sector from -25° to 25° .

The graphs of the horizontal measurements appear slightly asymmetrical, an effect which is investigated in the following.

A possible explanation for the asymmetry is the sensor mount, which protrudes out on the right side of the array, attenuating the signal received by antennas on that side when the reflector is behind said protrusion [NEEDS DRAWING].

Investigating the individual channel gains (c.f. 3.12) gives a first indication: the asymmetry is most pronounced when comparing the range of 30° to 60° to its counterpart of -30° to -60° for the horizontal measurements. Namely, in the former range, the gain seems to oscillate every 16 channels.

To explore further, figure 3.13 zooms in on one period. It shows the channel gain of the horizontal measurements for angles 30° to 60° of the channels in which transmit antenna 0 is active, arranged by the horizontal position of the receive antenna, from left to right. It can now be seen that the channel gain of antennas closer to the sensor mount's protrusion drops in amplitude at lower angles than that of the antennas further away from it, confirming that the sensor mount is responsible for the asymmetry. To get a more accurate estimate of the channel characteristics, the measurements have to be repeated with the sensor mounted on the outside.

With this, the individual antenna gains and the channel characteristics have been extracted from the data. In the next step, the recorded phases are analyzed.

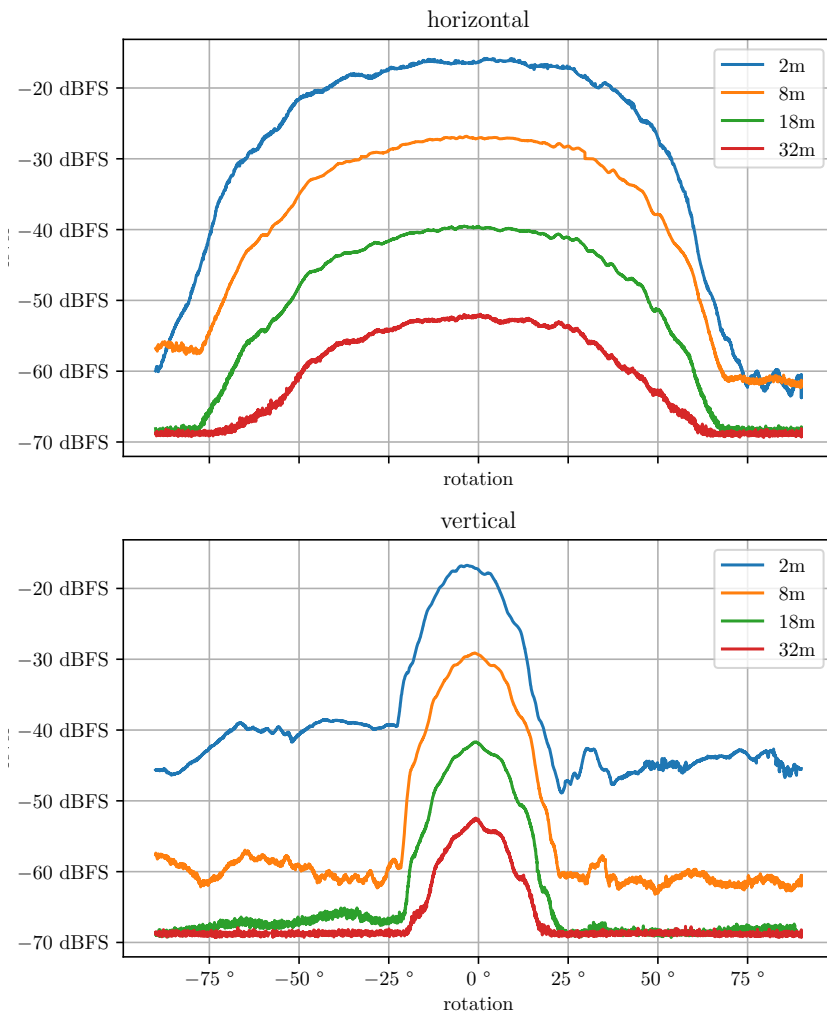


Figure 3.11: Mean Amplitude for multiple reflector distances

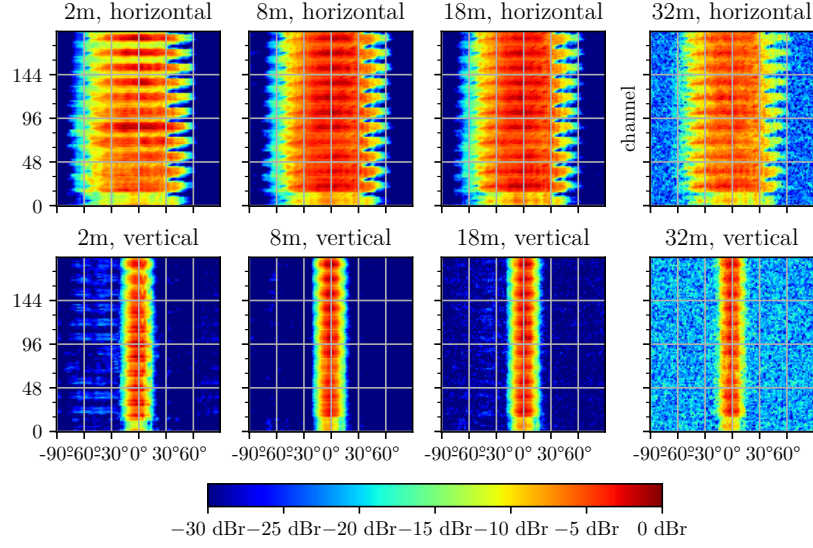


Figure 3.12: Channel-wise amplitude for multiple distances

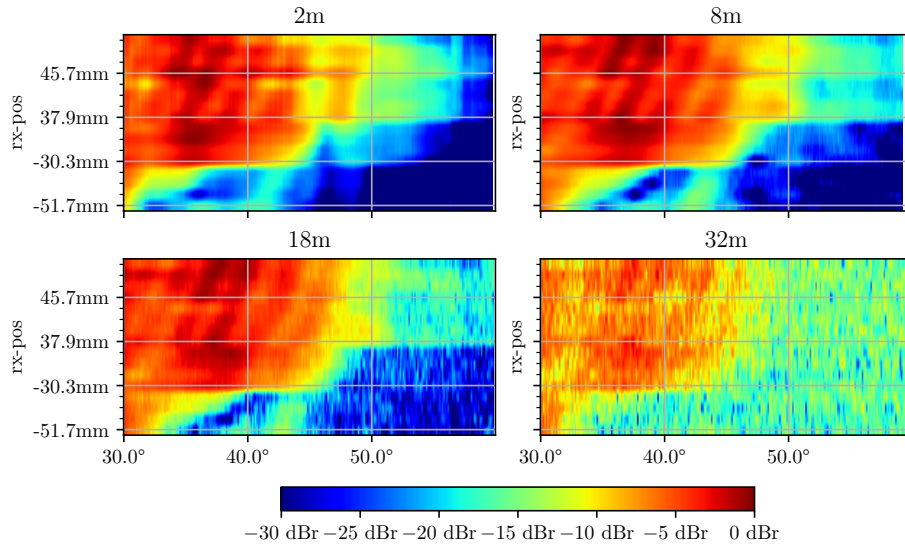


Figure 3.13: Channel-wise amplitude for multiple distances, channels arranged by Rx antenna position, only Tx antenna 0 active

3.3.5 Phase Estimation Techniques

The first step in extracting the phase parameters from the recorded data is yet again evaluating the signal's DFT phase at the optimum range bin $\hat{\Omega} = \frac{N\dot{\omega}}{c_0 f_s} \hat{R}_k$, which according to (3.7) should ideally yield

$$\Phi_k(\vec{r}_S) := \arg \mathcal{F}_m \{y_k[m]\}(\hat{\Omega}(\vec{r}_S)) \quad (3.18)$$

$$= \arg G_k(\vec{r}_S) \quad (3.19)$$

$$= \omega_0 \tau_k(\vec{r}_S) + \varphi_k \quad (3.20)$$

$$\text{with } \varphi_k = \varphi_{Tx,i} + \varphi_{Rx,j}, k = N_{rx}i + j \quad (3.21)$$

$$\text{and } \tau_k = \frac{2R_k}{c_0} \quad (3.22)$$

However, this part of the analysis is highly dependent on the accuracy of our assumptions, specifically the distance between reflector and sensor. The phase $\Phi_k(\theta) = \omega_0 \frac{2R_k(\theta)}{c_0} + \varphi_k$ changes immensely with small changes in said distance.

To illustrate, a change of $\Delta R_k = 1 \text{ mm}$ would cause approximately the following phase shift:

$$\begin{aligned} \Delta \Phi_k &= \omega_0 \frac{2\Delta R_k}{c_0} \\ &= 2\pi \cdot 77 \cdot 10^9 \text{ Hz} \cdot \frac{2 \cdot 1 \cdot 10^{-3} \text{ m}}{3 \cdot 10^8 \text{ m/s}} \\ &= 0.5133\pi = 92.4^\circ \end{aligned}$$

The range estimate made in section 3.3.3 had an RMSE of at least 11 mm, which introduces high uncertainty into the phase evaluation.

Another method to extract the phase from the recorded data is evaluating the FFT at the spectral peak, which was the method of choice in 3.2. In the following, we will illustrate the effects of interference with a simulated signal, as well as the impact of the interference on the two estimation methods accuracy.

Consider the signal received by an arbitrary channel of the array $y(t, \theta)$. Its frequency slowly changes as the orientation angle θ ; for simplicity's sake, let the frequency shift linearly with $\omega(\theta) = 2\pi(5 \text{ MHz} + 1 \text{ MHz} \cdot \theta)$, $\theta \in [-1, +3]$. This results in our ideal signal

$$y_{id}(t = mT_s, \theta) = 1 \cdot e^{-j\omega(\theta)mT_s}$$

Introducing a source of interference with constant frequency ω_0 and the amplitude 0.1, as well as a gaussian noise source $n(t, \theta) \sim \mathcal{N}(0; 0.001)$, we get

TODO: remove $n(t, \theta)$; just use vmin in image plots

$$y(t = mT_s, \theta) = 1 \cdot e^{-j\omega(\theta)mT_s} + 0.1 \cdot e^{-j\omega_0 mT_s} + n(t, \theta)$$

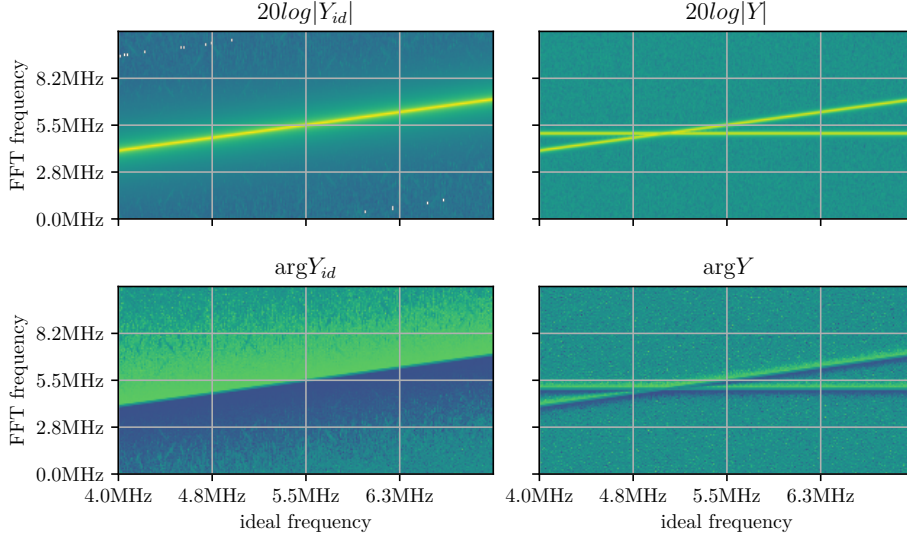


Figure 3.14: Spectra of example signal Y and Y_{id}

The spectrum of the ideal signal $Y_{id}(\Omega, \theta)$ and the degraded signal $Y(\Omega, \theta)$, calculated with an $N = 2048$ point FFT with $M = 512$ samples for every one of the 1000 “angles” θ , is shown in fig. 3.14 in amplitude and phase. The location of the both signals’ spectral maxima can be seen in fig. 3.15a, showing that the interference causes the spectral maximum to move from its ideal location. Figure 3.15b finally shows the effect of interference on the phase. We compare the phase of the ideal signal to phase of the degraded signal, which is evaluated at both the ideal and the degraded signal’s spectral maxima. It can be seen that the latter method’s phase diverges much more strongly from the ideal phase than that of the former, and should therefore be preferred in the subsequent analysis.

3.3.6 Phase Analysis

With knowledge of the robustness of our phase estimation against range estimation errors, the task of extracting the phase parameters can be continued. For better legibility of any graphs generated from the signal, the phase of any given channel will be unwrapped from its original domain of $\theta \in [-\pi + \theta_s, \pi + \theta_s]$ such that no discontinuities are visible when plotting phase against sensor orientation, while maintaining the original phase for $\theta = \theta_s$.

To assess the performance of the two techniques introduced in the previous section, figure 3.16 shows how they each compare to the signal model ¹ for a few selected channels. As it turns out, while being theoretically less robust

¹In this case, the signal model shown for comparison assumes that $\varphi_k = 0, \forall 0 \leq k < K$.

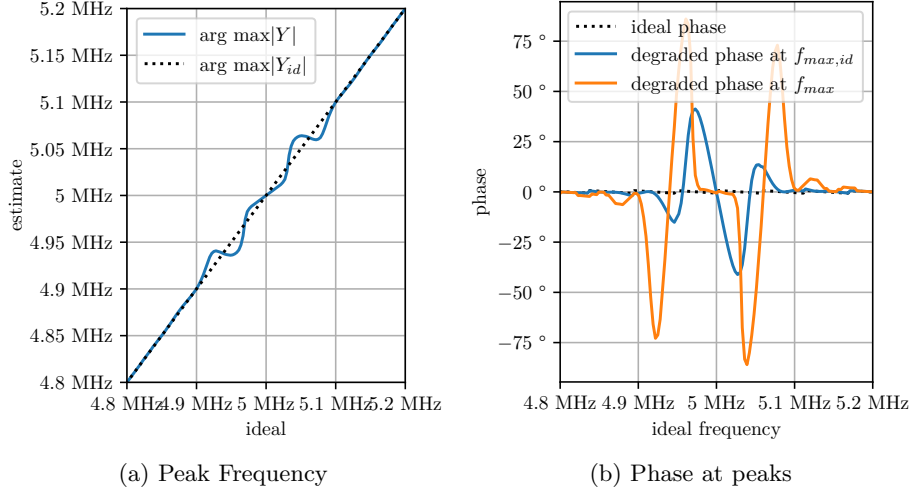


Figure 3.15: Extracting Phase from example spectra Y and Y_{id}

against interference, the phase evaluated at the signal peak is much smoother and matches our signal model much more closely.

Now, the channel phase offsets $\hat{\varphi}_k$ can be estimated using linear regression. Figure 3.17 shows an example linear regression, and the resulting channel phase offsets are shown in figure 3.18.

3.3.7 Antenna Separation

To describe the array in terms of its channel gains is somewhat redundant. As was defined in the underlying signal model, each channel gain $C_k e^{j\varphi_k}$ in actuality just the product of the transmit and receive antenna gains, that is

$$C_k e^{j\varphi_k} = A_i e^{j\psi_i} \cdot B_j e^{j\vartheta_j} \text{ for } k = N_{Tx}i + j \quad (3.23)$$

To more accurately describe the array, the next step in our analysis is to estimate antenna gains of the receive and transmit array. Since multiple combinations are feasible, the rightmost receive antenna 0 is defined as reference antenna, such that $B_0 = 1$ and $\vartheta_0 = 0$.

In the following, the channel gain amplitudes \hat{C}_k that were estimated in section 3.3.4, and the channel phase offsets $\hat{\varphi}_k$ from section 3.3.6 are used to obtain estimates for the transmit and receive antenna parameters.

Thus, the phase shown here is directly proportional to the estimated range \hat{R}_k :

$$\hat{\Phi}_k(\theta) = \omega_0 \tau_k + 0 = \frac{\omega_0}{c_0} 2\hat{R}_k(\theta).$$

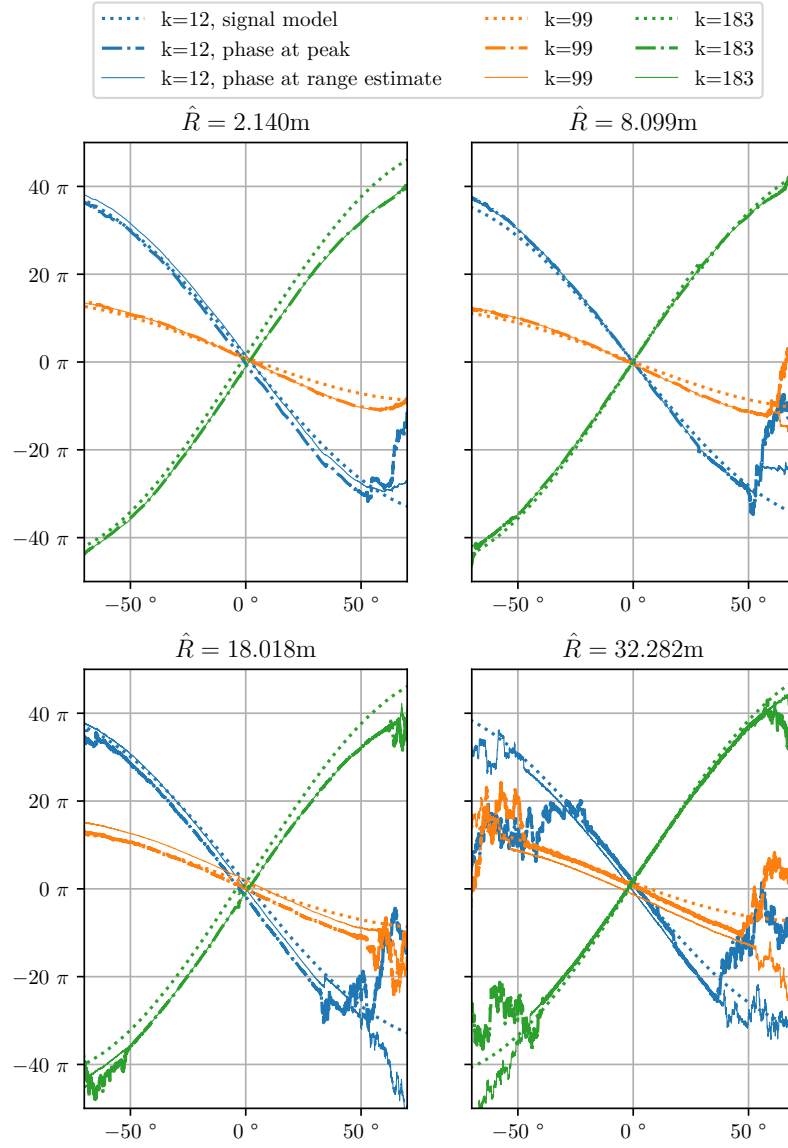


Figure 3.16: Extracted Phase from measured signal, compared to signal model

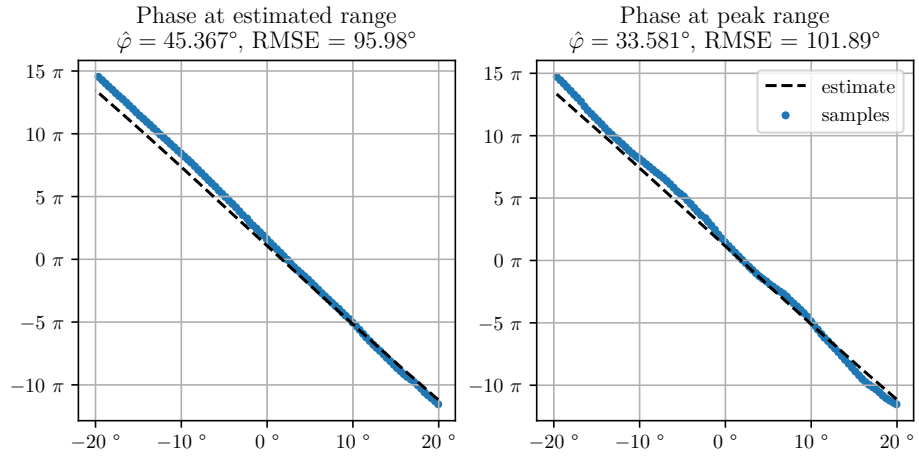


Figure 3.17: Example: Channel 12 Phase Linear Regression

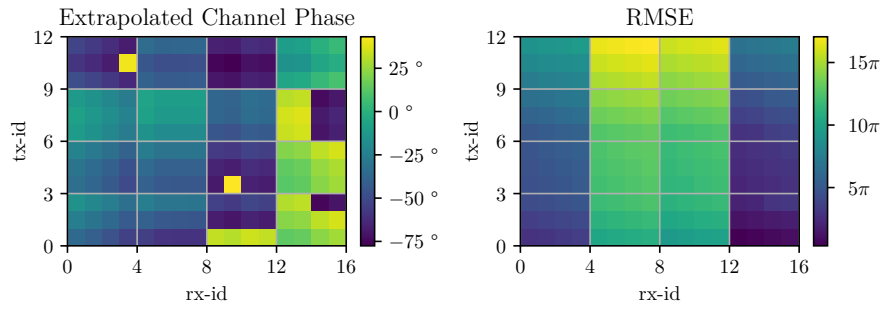


Figure 3.18: Phase Linear Regression Overview

Thanks to the definition of a reference antenna, the transmit antennas' amplitude and phase are easily obtained:

$$\hat{A}_i = \hat{C}_k \Big|_{k=N_{Tx}i} \quad (3.24)$$

$$\hat{\psi}_i = \hat{\varphi}_k \Big|_{k=N_{Tx}i} \quad (3.25)$$

With that, a least squares estimate for the remaining receive antennas' amplitude and phase is:

$$\hat{B}_{j \neq 0} = \frac{1}{N_{Tx-1}} \sum_{i=1}^{N_{Tx-1}} \frac{\hat{C}_k}{\hat{A}_i} \Big|_{k=N_{Tx}i}, \quad (3.26)$$

$$\hat{\vartheta}_{j \neq 0} = \frac{1}{N_{Tx-1}} \sum_{i=1}^{N_{Tx-1}} \frac{\hat{\varphi}_k}{\hat{\psi}_i} \Big|_{k=N_{Tx}i} \quad (3.27)$$

The resulting antenna parameters are shown in tables 3.1a and 3.1a.

			Antenna	Gain	Phase
			0	1.000	0.000°
Antenna	Gain	Phase	1	1.208	-7.095°
0	0.638	53.596°	2	1.294	-26.095°
1	0.879	125.805°	3	1.193	0.712°
2	0.852	48.354°	4	1.828	-1.052°
3	0.780	-126.556°	5	1.790	26.672°
4	0.732	-13.976°	6	1.881	42.318°
5	0.819	-111.434°	7	1.793	-2.503°
6	0.714	-81.855°	8	1.752	49.492°
7	0.625	82.760°	9	1.605	-2.698°
8	0.589	119.322°	10	1.566	69.663°
9	0.528	1.485°	11	1.533	-40.625°
10	0.475	7.703°	12	1.178	8.481°
11	0.461	-53.846°	13	0.992	-11.289°
(a) Tx Antennas			14	1.061	-1.515°
			15	0.904	-5.109°
			(b) Rx Antenna Gains		

Table 3.1: Estimated Antenna Parameters with Reference Rx0

TODO: include Formelzeichen A, B, ψ, ϑ

3.3.8 Conclusion

In this section, a method for estimating the antenna gains from real-world antenna measurements has been formulated that optimizes the estimation's accuracy and robustness against interference. The summarized steps are as follows:

1. Setup the reflector in front of the sensor. For each distance to measure, center the reflector by observing a live readout of the phase at the spectral peak caused by the reflector, and positioning the sensor such that the phase differences between channels is minimal.
2. From the measurements, extract the range of the spectral peak caused by the reflector for each channel and sensor orientation. Use numerical optimization to find a set of model parameters (\hat{R}_s , $\hat{\theta}_s$ or $\hat{\phi}_s$, and $\hat{\epsilon}$) that best matches the spectral peaks.
3. Evaluate the amplitude of the spectrum at the frequency corresponding to the estimated range from the previous step. The maximum amplitude across all angles is the channel gain C_k/R^2 , and the remaining attenuation is the channel characteristic $\hat{c}_k(\theta, \phi) \in [0, 1]$. With linear regression across all measured distances, the channel gain at 1 m can be estimated. The estimated channel characteristic $\hat{c}_k(\theta, \phi)$ can indicate orientations with low SNR and/or high interference.
4. Evaluate the phase of the spectrum at the frequency corresponding to the estimated range. Unwrap the phase of each channel such that the phase for $\theta = \hat{\theta}_s$ is in $[-\pi, \pi]$. Find the channel phase offset ϕ_k that best matches the signal model to the measured phase.
5. Separate each channel gain $\hat{C}_k e^{j\varphi_k}$ into a transmit gain $A_i e^{j\psi_i}$ and a receive gain $B_j e^{j\vartheta_j}$. Do this by defining a receive antenna as a reference.

Following these steps gives a decent estimate of the systems's physical properties with relatively lax requirements on the measurement itself. This is especially desirable since section 3.2 demonstrated a need for frequent in-field calibration, since the system parameters may change after restarting, and also shift over time.

More sophisticated methodology is required to achieve accurate measurements of the system. Controlling the positioning of the reflector relative to the sensor more closely, for example by placing the reflector on a linear axis, would greatly improve accuracy.

Interference sources in the environment would also have to be reduced. This would involve moving the experiment to a more controlled environment, such as a low-reflection room.

Another area of improvement lies in channel separation. Since the reference antenna in the described method is not external, but rather a part of the system under test, all the results of our analysis indicate are relative differences between antennas. Instead, an external antenna with known characteristics could replace the reflector, allowing for more accurate measurements of the individual antennas' absolute characteristics.

Chapter 4

Image Reconstruction

- recap results of last chapter - restate motivation of imaging - chapter overview

4.1 PyTorch

The algorithms implemented in this chapter are presented in the PyTorch framework. As a machine learning framework, it provides accelerated tensor processing. Tensors, ie. rectangular multidimensional arrays are a perfect fit to describe the signals sent by the sensor and the subsequent signal processing.

Since a huge part of the operations in the imaging algorithms are parallelizable, substantial performance gains can be achieved with the built-in parallel processing of tensor operations in PyTorch. While parallel processing on a CPU already substantially improves runtimes, even better performance can be achieved with GPU-enabled parallel processing, which is currently available using the CUDA-API on NVIDIA GPUs.

To aid in understanding any code listings referenced in this chapter, refer to a brief tutorial in section A.1 that discusses some key parts of PyTorch's syntax for tensor processing.

4.2 FFT-Based Imaging

The first imaging algorithm to be implemented is based on the FFT. A detailed description of the algorithm can be found in section 2.3.1, but the following is an abridged summary to reiterate the key principles.

The range is estimated by computing the FFT spectrum of each time signal; due to the FMCW principle, the frequency of the received signal is directly related to its range. Then, making the assumption of planar incident waves as well as having calibrated the array to achieve coherence between channels, the angle of arrival can be estimated using FFTs over the ULA subset of the sensor's virtual antenna array. The resolution can be increased by zero-padding

the signal before applying the FFT. An example implementation in PyTorch is shown in code listing A.1.

The function `calc_image` returns a 3D-image of dimension $N_{range} \times N_{azimuth} \times N_{elevation}$. Its input consists of `data`, a time data tensor of dimension $M \times K$, the calibration weights `weights` of dimension K , and `settings`, a dictionary of settings. `settings` is expected to contain the input and output dimensions, the indices of the ULA subset of the virtual array with gaps indicated by -1 , as well as the window functions to be used in each FFT.

4.2.1 Offline Calibration

To compute the angle of arrival, the input signal has to be made coherent in phase. This is done by deviding each channel's range spectrum $Y_k(\Omega)$ by the estimate channel gain $\hat{C}_k e^{j\hat{\varphi}_k}$. As well as aligning the phases correctly, this also equalizes the channel gains.

4.2.2 Direction of Arrival Accuracy

This section aims to illustrate the theoretical capabilities of FFT-based DoA-estimation on the iMCR. The directivity in azimuth and elevation is simulated for multiple distances to illustrate the impact of near-field conditions.

For this, the input time signal is an ideal point source (see eqn. 2.9), located at $(R, \theta = 0, \phi = 0)$ assuming a constant antenna gain of $\underline{G}_k(r, 0, 0) = 1 \forall r, k$:

$$\underline{y}_k[m] = e^{-j\hat{\omega} \frac{2R}{c_0} mT_s}, k \in [0, K-1], m \in [0, M-1] \quad (4.1)$$

The image is computed for multiple distances at resolution $N_{range} = 1024$, $N_{azimuth} = 2048$ and $N_{elevation} = 8$. and evaluated at the range and evaluation bin of the scatterer. The resulting peaks in azimuth are shown in figure 4.1.

The Fraunhofer distance of the array can be computed from the wavelength $\lambda_0 = 3.944 \text{ mm}$ and the virtual array's azimuth aperture $D_{azm} = 86 \frac{\lambda_0}{2} = 16.96 \text{ cm}$:

$$d_f = \frac{2D_{azm}^2}{\lambda_0} = 14.59 \text{ m} \quad (4.2)$$

It can be seen that for targets closer than d_f , the peak deteriorates in multiple ways. The location of the peak moves to higher angles than the actual location of $\theta = 0^\circ$. For targets at 30 cm, the peak has moved to $\theta = 2.02^\circ$.

In near-field conditions, the side lobe levels increase until their minimum only around 20 dB below the peak, whereas in the far field, the minimum is 60 dB below the peak.

In far-field conditions, the main peak at $\theta = 0^\circ$ is clearly separated from its sidelobes by pronounced local minima. The minimum to the right of the

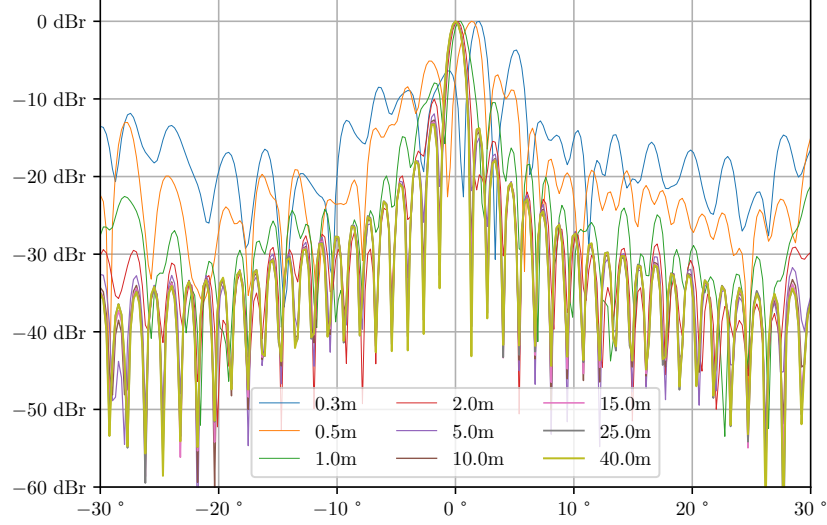


Figure 4.1: iMCR FFT direction of arrival estimation: theoretical accuracy in azimuth for an ideal point scatterer at different ranges

main peak starts to disappear below d_f , and the peak to the right starts to merge with the main peak the closer the target gets. The far-field half-power and quarter-power beamwidths are approximately $\theta_{3\text{dB}} = 1.9^\circ$ and $\theta_{6\text{dB}} = 2.6^\circ$, respectively.

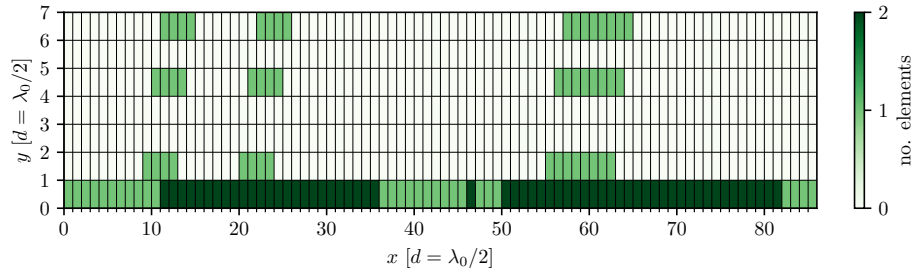


Figure 4.2: iMCR virtual antenna array occupancy

For direction of arrival estimation in azimuth, the 86-channel ULA subset of the array is well applicable to an FFT-based approach, even with its reduced performance in near-field conditions. Yet in elevation, such a ULA subset does not exist, as the array is sparse horizontally and vertically (c.f. 4.2).

To still use the FFT, an approach would be “backfilling” the array with zeros to obtain a complete uniformly spaced rectangular virtual array. The FFT is then computed over the columns (c.f. 4.2) of this array, each yielding a spectrum that is related to the direction of arrival in elevation.

A disadvantage of this approach is that it can cause substantial spectral leakage. Our backfilled array $b[k]$ is mathematically equivalent to applying a binary window function $w[k]$ to the ideal rectangular array $a[k]$:

$$\begin{aligned} b[k] &= w[k] \cdot a[k] \\ \text{e.g. } b[0..7] &= [b_0, b_1, 0, 0, b_2, 0, b_3, 0], \\ w[0..7] &= [1, 1, 0, 0, 1, 0, 1, 0], \\ \text{and } a[0..7] &= [a_0, a_1, a_2, \dots, a_7] \end{aligned}$$

However, this window function causes additional peaks to appear in the spectrum, since multiplication in the time domain is equivalent to convolution in the frequency domain:

$$\begin{aligned} \mathcal{F}\{b[k]\}[n] &= \mathcal{F}\{w[k] \cdot a[k]\}[n] \\ &= W[n] * A[n] \end{aligned}$$

In case of the example given above,

$$|W[0..7]| = [4.00, 0.77, 1.41, 1.85, 2.00, 1.85, 1.41, 0.77]$$

It would be possible to reduce the spectral leakage with deconvolution algorithms. To show the starting point for these algorithms, the unadulterated spectral peak for targets at multiple distances is shown in figure 4.3.

It can be seen that the half-power beamwidth improves the more channels are included in the sub-array: for two, three and four active channels, we approximately get $\phi_{3\text{dB}} = 60^\circ$, 14° and 12° , respectively.

As the peak narrows, the spectrum contains more and more sidelobes. For sub-arrays that include three channels, the side peaks are the strongest, barely 3 dB less than the main peak.

The Fraunhofer distance of the array in elevation can be computed from the wavelength $\lambda_0 = 3.944\text{ mm}$ and the virtual array’s maximal elevation aperture $D_{elv} = 7\frac{\lambda_0}{2} = 1.38\text{ cm}$:

$$d_f = \frac{2D_{elv}^2}{\lambda_0} = 9.66\text{ cm} \quad (4.3)$$

The Fraunhofer distance in elevation is smaller than in azimuth by an order of magnitude. Thus, all distances considered lie in the far-field, and the shape of the peak should not change a lot over distance.

While this holds true for the two- and three-channel case, if four channels are considered, the shape does change with distance.

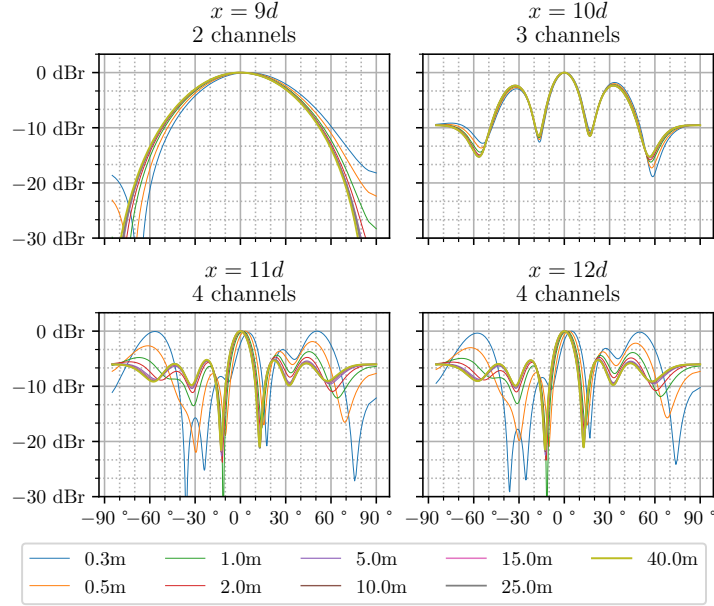


Figure 4.3: iMCR FFT-based direction of arrival: theoretical accuracy in elevation for an ideal point scatterer at different ranges and different array subsets

On the one hand, the location of the main peak moves to higher angles if the range is decreased. On the other hand, the first two sidelobes decrease in amplitude, while the third ones move towards $\phi = 0^\circ$ and increase in amplitude.

TODO:

- Change Code listing
- Mention Taper Functions!!
- Conclude

4.2.3 Result

- BVD tower image
- maybe 3d image?
- Runtime, real-time capabilities

4.3 Backprojection Imaging

The second imaging algorithm to be implemented is backprojection. A detailed description of the algorithm can be found in section 2.3.2, but the gist of it is the following:

Backprojection works by computing the correlation between the incoming time data and the signal of an ideal point scatterer (see eqn. 2.9) at an ar-

bitrary set of locations. The signal uses the measured antenna gains at that location. Assuming the antenna gains to be static, the ideal signals can be pre-computed to be used to generate images for multiple measurements. An example implementation in PyTorch is shown in code listing A.3.

TODO:

- how to calculate BP weights from measured antenna gains

A major limitation of the way the backprojection algorithm is currently implemented is its scalability. Pre-calculating the weights, while allowing for decent opportunities in decreasing the runtime through parallel processing, quickly use a prohibitive amount of memory.

If there are Z individual locations, M sample points in time and K channels, then $M \cdot K \cdot Z$ weights need to be computed. For example, the weights for a uniformly spaced image with a rather low resolution of $Z = 100 \cdot 10 \cdot 100 = 100$ kilovoxels, at $K = 192$ and $M = 1022$, stored as complex numbers consisting of two 64 bit floating point numbers, would require

$$\begin{aligned} & M \cdot K \cdot Z \cdot 2 \cdot 64 \text{ bit} \\ &= 1022 \cdot 192 \cdot 10^5 \cdot 16 \text{ byte} \\ &= 3.14 \cdot 10^{12} \text{ byte} = 3.14 \text{ TB} \end{aligned}$$

of memory. With the projections of Moore’s law [citation needed], this amount of memory should be available in consumer-level computers by the year 2050. Until then, the memory requirements has to be reduced by re-computing the weights and applying them to the input data along an arbitrary dimension. For example, recomputing them for every of the sampled time points would reduce the memory requirements by a factor of 1022 to only around 3 GB.

4.3.1 Direction of Arrival Accuracy

To better understand the DoA-estimation capabilities of the algorithm, the azimuth and elevation is simulated for multiple distances to illustrate the impact of near-field conditions. The same input signal as in section 4.2.2 is used.

The image is computed for multiple distances at resolution $N_{range} = 1$, $N_{azimuth} = 512$ and $N_{elevation} = 1$. Naturally, the range of the image is set to the range of the ideal target each time. The resulting peaks in azimuth and elevation are shown in figures 4.4 and 4.5, respectively.

In both azimuth and elevation, it can be seen that the peak no longer deteriorates in near-field conditions. The peak and most side lobes remain virtually identical across all considered ranges. In azimuth, the level of the sidelobes stabilizes at around 30 dB to 40 dB below the peak, while in elevation, the sidelobe level is merely 3 dB to 6 dB below the peak.

In azimuth, the half-power beamwidth is approximately $\theta_{3\text{dB}} = 1.6^\circ$ and the quarter-power beamwidth approximately $\theta_{6\text{dB}} = 2.4^\circ$. In elevation, the

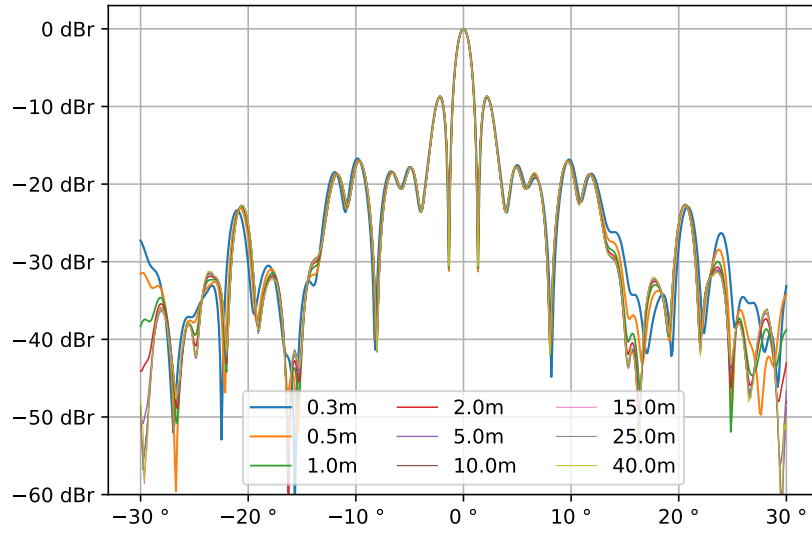


Figure 4.4: iMCR backprojection direction of arrival estimation: theoretical accuracy in azimuth for an ideal point scatterer at different ranges

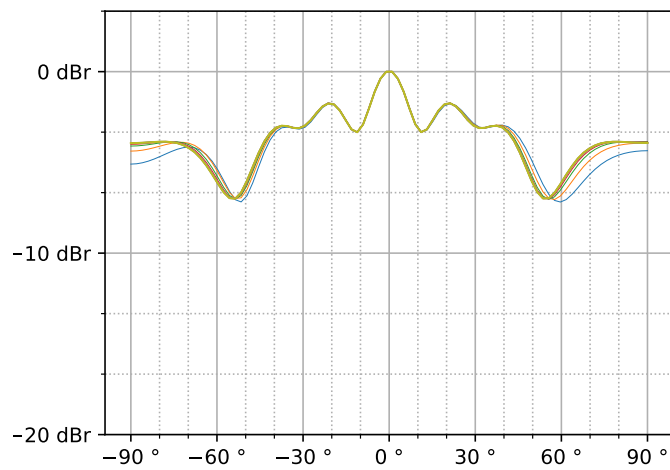


Figure 4.5: iMCR backprojection direction of arrival estimation: theoretical accuracy in elevation for an ideal point scatterer at different ranges

half-power beamwidth is approximately $\phi_{3\text{ dB}} = 20^\circ$ and the quarter-power beamwidth approximately $\phi_{6\text{ dB}} = 100^\circ$.

TODO:

- discuss windowing
- discuss elevation/azimuth trade-off: should elevation-channels contribute more?

4.3.2 Result

4.4 Hybrid Approach

- Runtime-Performance Trade-Off -

4.4.1 Direction of Arrival Accuracy

- Phase error due to frequency quantization
 - Plot: near-field beampattern for different FFT-lengths

4.5 Conclusion

Chapter 5

Conclusions

Chapter 6

Discussion and Outlook

Appendix A

Code

A.1 Introduction to PyTorch Sensor Processing

Pytorch Tensors can be created from python lists, or with a utility function like `zeros()`, `ones()` or `full()`:

```
1 >>> import torch
2 >>> torch.tensor(list(range(2)))
3 tensor([0, 1])
4 >>> torch.ones((2,3))
5 tensor([[1., 1., 1.],
6         [1., 1., 1.]])
```

Python's basic binary operators, such as `+`, `-`, `*` or `/`, can be used to operate on tensors. If both tensors are the same size, the operation is done element by element, similar to the hadamard product of matrices. Scalars can also be applied to tensors.

```
1 >>> torch.tensor(list(range(2))) + torch.ones(2)
2 tensor([1., 2.])
3 >>> 5 + torch.zeros(2)
4 tensor([5., 5.] )
```

These operations are also allowed for tensors of different size, if their shapes follow the requirement that “[...] iterating over the dimension sizes, starting at the trailing dimension, the dimension sizes must either be equal, one of them is 1, or one of them does not exist.”¹

¹see <https://pytorch.org/docs/stable/notes/broadcasting.html>

For example, a $3 \times 2 \times 1$ tensor and a 1×2 tensor are broadcastable:

```
1 >>> a = torch.tensor([[[0],[1]],[[2],[3]],[[4],[5]]])
2 >>> b = torch.tensor([1,10])
3 >>> a.shape, b.shape
4 (torch.Size([3, 2, 1]), torch.Size([1, 2]))
5 >>> a+b
6 tensor([[[ 1, 10],
7          [ 2, 11]],
8
9          [[ 3, 12],
10         [ 4, 13]],
11
12         [[ 5, 14],
13         [ 6, 15]]])
```

In the third dimension, the elements of tensor **a** are broadcast onto the elements of tensor **b**. In the second dimension, the inverse is the case. The resulting tensor has the shape $3 \times 2 \times 2$.

Tensor indexing of one-dimensional tensors is identical to list indexing in python; and multi-dimensional tensors are indexed with a tuple of indices or slices. For example, to select all first elements of the previous result's third dimension, the following syntax is used:

```
1 >>> c=a+b
2 >>> c[:, :, 0]
3 tensor([1, 2],
4         [3, 4],
5         [5, 6])
```

Dimensions where a single index was used are omitted from the result. It is also possible to insert dimensions with the keyword **None**:

```
1 >>> c[:, :, 0].shape
2 torch.Size([3, 2])
3 >>> c[:, None, :, 0].shape
4 torch.Size([3, 1, 2])
```

Besides these basic operations, a plethora of functions is already implemented for tensors. Often, for each function operating on a tensor, there is also an equivalent member function of the class tensor that does the same. This makes composing functions a lot easier, because no nested parentheses are required:

```
1 >>> a = torch.randn((1,100))
2 >>> 20*torch.log10(torch.abs(torch.mean(a)))
3 tensor(-16.3953)
4 >>> 20*a.mean().abs().log10()
5 tensor(-16.3953)
```

A.2 Listings

```
1 def calc_image_fft(data, calibration, settings):
2     # range fft
3     W,N = settings['range_window'], settings['range_fftlens']
4     range_data = window_fft(data,W,N,0)
5     range_data /= calibration[None,:] # apply calibration
6
7     # azimuth fft
8     idx,W,N = settings['azm_ula'], settings['azm_window'], settings['azm_fftlens']
9     azimuth = window_fft(range_data[:,idx],W,N,1)
10
11     # elevation ffts
12     idx,W = settings['elv_ula'], settings['elv_window']
13     M,N,L = settings['range_fftlens'], settings['elv_fftlens'], idx.shape[-1]
14     elevation = torch.empty((M,N,L), dtype=torch.cfloat)
15     for l in range(L):
16         elevation[:, :, l] = window_fft(range_data[:,idx[:,l]],W,N,1)
17
18     # estimate elevation gain
19     elevation_est = elevation.abs().mean(-1)
20     elevation_est /= elevation_est.max()
21
22     # return 3d image
23     img_3d = azimuth[:, :, None] * elevation_est[:, None, :]
24     return img_3d
```

Listing A.1: PyTorch implementation of the FFT-based imaging algorithm

```
1 def window_fft(x, window, fftlen, dim):
2     fft = torch.fft.fft(x.transpose(dim,-1)*window, fftlen)
3     fft = torch.fft.fftshift(fft)/window.sum()
4     return fft.transpose(dim,-1)
```

Listing A.2: PyTorch computation of a normalized FFT spectrum using a window function

```

1 def calc_image_bp(data, settings, pos):
2     M,K,L = data.shape
3     PP = pos.shape
4
5     gain = settings['channel gain']                # PP x K
6     slope = settings['chirp slope']
7     f0 = settings['start frequency']
8     Ts = settings['sample period']
9
10    # Compute weights
11    tau = time_of_flight(settings, pos).unsqueeze(-2) # PP x 1 x K
12    t = Ts*torch.arange(M)[: ,None]                 # M x 1
13
14    freq = 2*torch.pi*slope*tau
15    phase = 2*torch.pi*f0*tau
16    weights = gain.unsqueeze(-2) * \
17        torch.exp(1j*freq*t) * \
18        torch.exp(1j*phase) # PP x M x K
19
20    # Compute 3D Images
21    imgs = torch.empty((*PP,L), dtype=torch.cfloat)
22    for l in range(L):
23        imgs[... ,l] = (weights.conj() * data[:, :, l]).mean((-2,-1))
24    return imgs

```

Listing A.3: PyTorch implementation of the backprojection imaging algorithm

```

1 def time_of_flight(settings, pos):
2     x_tx, x_rx = settings['x_tx'], settings['x_rx']
3     y_tx, y_rx = settings['y_tx'], settings['y_rx']
4     K=len(x_tx)
5     c0 = settings['lightspeed']
6
7     txpos = torch.tensor([x_tx,y_tx,[0]*K]).transpose(0,1)
8     rxpos = torch.tensor([x_rx,y_rx,[0]*K]).transpose(0,1)
9     # K x 3
10    pos = pos.unsqueeze(-2)
11    # PP x 1 x 3
12    r_tx = (txpos-pos).square().sum(-1).sqrt()
13    r_rx = (rxpos-pos).square().sum(-1).sqrt()
14    # sum(PPx1x3 - Kx3, -1) = PPxK
15    return (r_tx+r_rx)/c0

```

Listing A.4: PyTorch computation of the channel-wise time of flight for a given position

```

1 def calc_pos(settings):
2     if settings['coordinate type'] == 'cartesian':
3         x0,x1,y0,y1,z0,z1 = settings['limits']
4         X,Y,Z = settings['resolution']
5         x = torch.linspace(x0,x1,X)[: ,None,None].expand(-1,Y,Z)
6         y = torch.linspace(y0,y1,Y)[None, :,None].expand(X,-1,Z)
7         z = torch.linspace(z0,z1,Z)[None,None, :].expand(X,Y,-1)
8     elif settings['coordinate type'] == 'cylindrical':
9         r0,r1,theta0,theta1,z0,z1 = settings['limits']
10        R,Theta,Z = settings['resolution']
11        r = torch.linspace(r0,r1,R)
12        theta = torch.linspace(theta0,theta1,Theta)
13        x = (r[: ,None,None] * theta.cos()[None, :,None]).expand(-1,-1,Z)
14        y = (r[: ,None,None] * theta.sin()[None, :,None]).expand(-1,-1,Z)
15        z = torch.linspace(z0,z1,Z)[None,None, :].expand(R,Theta,-1)
16    elif settings['coordinate type'] == 'spherical':
17        r0,r1,theta0,theta1,phi0,phi1 = settings['limits']
18        R,Theta,Phi = settings['resolution']
19        r = torch.linspace(r0,r1,R)
20        theta = torch.linspace(theta0,theta1,Theta)
21        phi = torch.linspace(phi0,phi1,Phi)
22        x = r[: ,None,None] * theta[None, :,None].cos() * phi[None,None, :].sin()
23        y = r[: ,None,None] * theta[None, :,None].sin() * phi[None,None, :].sin()
24        z = (r[: ,None,None] * phi[None, :,None].cos()).expand(-1,Theta,-1)
25
26    return torch.stack((x,y,z),-1)

```

Listing A.5: PyTorch computation of uniformly spaced sample positions in different coordinate systems

Appendix B

Additional Experiments

List of Figures

3.1	Measurement Setup	17
3.2	Measured System Temperature After Startup	17
3.3	Expected System Temperature over Time for Interrupted Power	18
3.4	Mean Intensity Spectrum	19
3.5	Change in reflector distance as measured by iMCR	20
3.6	Recorded drift over night	21
3.7	Recorded drift and temperature with system restart	22
3.8	Example Optimization Results	26
3.9	Example Channel Gain Linear Regression	28
3.10	Channel Gain Linear Regression Results	29
3.11	Mean Amplitude for multiple reflector distances	30
3.12	Channel-wise amplitude for multiple distances	31
3.13	Channel-wise amplitude for multiple distances, channels arranged by Rx antenna position, only Tx antenna 0 active	31
3.14	Spectra of example signal Y and Y_{id}	33
3.15	Extracting Phase from example spectra Y and Y_{id}	34
3.16	Extracted Phase from measured signal, compared to signal model	35
3.17	Example: Channel 12 Phase Linear Regression	36
3.18	Phase Linear Regression Overview	36
4.1	iMCR FFT direction of arrival estimation: theoretical accuracy in azimuth for an ideal point scatterer at different ranges	41
4.2	iMCR virtual antenna array occupancy	41
4.3	iMCR FFT-based direction of arrival: theoretical accuracy in ele- vation for an ideal point scatterer at different ranges and different array subsets	43
4.4	iMCR backprojection direction of arrival estimation: theoretical accuracy in azimuth for an ideal point scatterer at different ranges	45
4.5	iMCR backprojection direction of arrival estimation: theoretical accuracy in elevation for an ideal point scatterer at different ranges	45

List of Tables

3.1	Estimated Antenna Parameters with Reference Rx0	37
-----	---	----